Éric Walter and Luc Pronzato

Identification of Parametric Models

from Experimental Data

Springer

Éric Walter and Luc Pronzato

Identification of Parametric Models

from Experimental Data

Translated from an updated French version
by the authors, with the help of

John Norton

School of Electronic and Electrical Engineering,
University of Birmingham, UK

With 119 Figures

Springer

MASSON

Paris Milan Barcelone

Éric Walter
Laboratoire des Signaux et Systèmes, CNRS–SUPÉLEC,
91192 Gif sur Yvette Cedex, France

Luc Pronzato
Laboratoire I3S, CNRS, Sophia-Antipolis,
06560 Valbonne, France

ISBN 3-540-76119-5 Springer-Verlag Berlin Heidelberg New York

British Library Cataloguing in Publication Data
Walter, E. (Eric)
Identification of parametric models : from experimental
data. - (Communications and control engineering series)
1. Parameter estimation 2. Mathematical models
I. Title II. Pronzato, Luc
519.5
ISBN 3540761195

Library of Congress Cataloging in Publication Data
A catalog record for this book is available from the Library of Congress

Printed in Great Britain

Published with the support of the Ministère de l'enseignement supérieure et de la recherche (MESR, France):
Direction de l'information scientifique et technique et des bibliothèques (DISTB)

Original French edition
Identification de modèles paramétriques à partir de données expérimentales
© Masson, Paris, 1994
Masson ISBN 2-225-85381-9
Masson, 120 boulevard Saint-Germain, 75280 Paris Cedex 06, France

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Typesetting: Camera ready by authors
Printed and bound at the Athenæum Press Ltd., Gateshead
69/3830-543210 Printed on acid-free paper

Preface

This book results from many years of research and teaching collaboration between the authors at the *Laboratoire des Signaux et Systèmes*, a laboratory common to the *Centre National de la Recherche Scientifique* (CNRS) and the *École Supérieure d'Électricité* (SUPÉLEC). It has evolved from notes for a graduate course at Paris-Sud University and a continuing education course for engineers at SUPÉLEC. It is thus aimed at two groups of readers.

The first consists of students wishing to familiarize themselves with the basic methods for system identification and parameter or state estimation. We have taken great pains to present methods in the most accessible way possible, and this book can be seen as an introduction to more sophisticated material. Exposition always moves from the simple to the more sophisticated, and many simple examples are discussed in detail. Numerous illustrations also facilitate understanding. The practical importance of the ideas is always stressed, and the limitations of the methods are explained.

The second possible readership consists of researchers or engineers who have to squeeze parameters out of experimental data. We should like to emphasise the interdisciplinary nature of this book, which should be of interest to people outside the Communications and Control Engineering communities proper. Practioners will get from it tools to analyze the quality of the estimates they produce, explanations of why their favourite software does not always yield the results they hope for, possible remedies for numerical difficulties and advice as to how to organize data collection. Within such a broad domain, researchers tend to specialize, so some may find the book useful for its breadth of coverage of techniques that may be of help in their future research. In this respect, they may find the large bibliography especially useful. Rather than providing detailed proofs of a limited number of technical results, we have chosen to introduce as many relevant notions and techniques as we could, explain why we are convinced of their importance, and indicate further reading.

Topics covered include choice of the structure of the mathematical model, choice of a performance criterion to compare models, optimization of this performance criterion, evaluation of the uncertainty in the estimated parameters, design of experiments and critical analysis of the results. Many recent methods are presented, some for the first time in a book on identification. Among the distinctive features of this volume are

— a presentation of the methodology for testing linear and nonlinear models for identifiability and distinguishability;

— an emphasis on other criteria than least squares (although the least-squares criterion is, of course, considered), and the importance of robustness;

— a detailed treatment of parametric optimization, including much more consideration of numerical aspects than usual (evaluation of the effect of rounding errors, generation of derivatives of the cost function with respect to the parameters by exact numerical methods, global optimization...); recursive and non-recursive methods are both considered, for models linear or nonlinear in their parameters;

— a description of deterministic and statistical methods of characterizing the uncertainty in the parameters resulting from that in the data;

— a much more detailed presentation of experiment design than usual.

We hope the many cross-references will help the reader navigate through the material, should he or she choose not to follow the order of the chapters.

Writing this book was made possible by the freedom and time given to us by CNRS and we are grateful to Pierre Bertrand, Head of the *Laboratoire des Signaux et Systèmes*, who provided us with particularly favourable working conditions.

Many ideas and references result from discussions we have had with our colleagues and students. May they forgive us for not always acknowledging it. We would especially like to thank John Happel, Luc Jaulin, Caroline Kulcsár, Gilles Le Cardinal, Yves Lecourtier, Hélène Piet-Lahanier, Alain Venot and Anatoli Zhigljavsky. This book would have been the poorer without their contributions.

We took the opportunity of the preparation of this English version to revise, update and expand the manuscript, so this is a second edition rather than merely a translation. Special thanks are due to John Norton who did much more than just improve our English. As could be feared, we were unable to refrain from introducing some last-minute corrections after his checking. Any remaining clumsiness should therefore be laid at our door.

Contents

Notation

Typography

Italic	scalar.
Bold lower case	column vector (row vectors are written as transposed column vectors).
Bold upper case	matrix.
Outlined upper case	set (*e.g.*, $\mathbb{R}$, $\mathbb{S}$).
$\mathbf{M}(s)$, $\mathbf{v}(s)$ and $v(s)$	Laplace transforms of $\mathbf{M}(t)$, $\mathbf{v}(t)$ and $v(t)$.
$\mathbf{M}^{-1}$	inverse of $\mathbf{M}$.
$\mathbf{M}^{\mathrm{T}}$ and $\mathbf{v}^{\mathrm{T}}$	transposed of $\mathbf{M}$ and $\mathbf{v}$.
$\Vert\mathbf{v}\Vert$	norm of $\mathbf{v}$.
$\Vert\mathbf{v}\Vert_2$	Euclidean norm of $\mathbf{v}$, equal to $(\mathbf{v}^{\mathrm{T}}\mathbf{v})^{1/2}$.

Common symbols and acronyms

$\mathbf{A}$	state matrix ($n_{\mathrm{x}} \times n_{\mathrm{x}}$).
$A(q, \mathbf{p})$	polynomial in q^{-1}, parametrized by $\mathbf{p} = (a_1, \ldots, a_{n_{\mathrm{a}}})^{\mathrm{T}}$.
AR	autoregressive.
ARARMAX	autoregressive with exogenous variable and ARMA noise.
ARARX	autoregressive with exogenous variable and AR noise.
$\arg\max j(\mathbf{p})$	value of $\mathbf{p}$ that maximizes j.
$\arg\min j(\mathbf{p})$	value of $\mathbf{p}$ that minimizes j.
ARMA	autoregressive with moving-average noise.
ARMAX	autoregressive with exogenous variable and MA noise.
ARX	autoregressive with exogenous variable.
$\mathbf{B}$	control matrix ($n_{\mathrm{x}} \times n_{\mathrm{u}}$).
$B(q, \mathbf{p})$	polynomial in q^{-1}, parametrized by $\mathbf{p} = (b_1, \ldots, b_{n_{\mathrm{b}}})^{\mathrm{T}}$.
$\mathbf{C}$	observation matrix ($n_{\mathrm{y}} \times n_{\mathrm{x}}$).
$C(q, \mathbf{p})$	polynomial in q^{-1}, parametrized by $\mathbf{p} = (c_1, \ldots, c_{n_{\mathrm{c}}})^{\mathrm{T}}$.
card $\mathbb{S}$	number of elements of set $\mathbb{S}$.
c.d.f.	cumulative distribution function.
$\mathbf{c}_{\mathrm{e}}(\mathbf{p})$	vector of equality constraints to be satisfied by $\mathbf{p}$ (written as $\mathbf{c}_{\mathrm{e}}(\mathbf{p}) = \mathbf{0}$).
$\hat{c}_{\mathrm{e}}(k)$	empirical autocorrelation of signal e.
$\mathbf{c}_{\mathrm{i}}(\mathbf{p})$	vector of inequality constraints to be satisfied by $\mathbf{p}$ (written as $\mathbf{c}_{\mathrm{i}}(\mathbf{p}) \leq \mathbf{0}$).
conv($\mathbb{S}$)	convex hull of set $\mathbb{S}$.

$\mathbf{d}$	search direction in parameter space.
$\mathbf{D}$	matrix describing the direct effect of the inputs on the outputs $(n_{\mathrm{y}} \times n_{\mathrm{u}})$.
$\mathbf{d}_{\mathbf{x}}$	dual variable of $\mathbf{x}$.
$D(q, \mathbf{p})$	polynomial in q^{-1}, parametrized by $\mathbf{p} = (d_1, \ \dots, d_{n_{\mathrm{d}}})^{\mathrm{T}}$.
$\det \mathbf{M}$	determinant of $\mathbf{M}$.
$\operatorname{diag} \mathbf{w}$	diagonal matrix, the ith diagonal entry of which is w_i.
$\dim \mathbf{v}$	dimension of $\mathbf{v}$.
$\partial j(\mathbf{p})$	subdifferential of j at $\mathbf{p}$.
$\mathbf{e}$ or e	error.
$\mathbb{E}(\hat{\mathbf{p}}, \mathbf{M})$	ellipsoid defined by $\{\mathbf{p} \in \mathbb{R}^{n_{\mathrm{p}}} \mid (\mathbf{p} - \hat{\mathbf{p}})^{\mathrm{T}}\mathbf{M}^{-1}(\mathbf{p} - \hat{\mathbf{p}}) \ \leq 1\}$.
$\underset{\mathbf{x}}{\mathrm{E}}\{.\}$ or $\mathrm{E}_{\mathbf{x}}\{.\}$	mathematical expectation with respect to $\mathbf{x}$.
e_{g}	generalized error.
e_{p}	prediction error.
$\mathbf{e}_{\mathrm{r}}$	regressor error.
e_{u}	input error.
e_{y}	output error.
$\mathbb{f}$	inclusion function associated with function $\mathbf{f}$.
$\overline{\mathbf{F}}$	average Fisher information matrix per unit of time.
$\mathcal{F}(n_1, n_2)$	Fisher-Snedecor distribution, with n_1 and n_2 degrees of freedom.
$\mathbf{F}(\mathbf{p})$ or $\mathbf{F}(\mathbf{p}, \Xi)$	Fisher information matrix.
$F(q, \mathbf{p})$	filter in the Box-Jenkins model.
$\mathbf{F}_{\mathrm{B}}(\mathbf{p}, \Xi)$	Bayesian information matrix.
$\mathbf{F}_{\mathrm{F}}$	Fisher information matrix associated with the parameters of $F(q, \mathbf{p})$ in the Box-Jenkins model.
$\mathbf{F}_{\mathrm{G}}$	Fisher information matrix associated with the parameters of $G(q, \mathbf{p})$ in the Box-Jenkins model.
FIR	Finite Impulse Response.
$\mathbf{F}_{\mathrm{ps}}(\mathbf{p})$	average Fisher information matrix per sample.
$F_{\alpha}(n_1, n_2)$	numerical value with a probability α of being exceeded by a random variable distributed $\mathcal{F}(n_1, n_2)$.
$\mathbf{g}(\mathbf{p})$	gradient of the cost function with respect to the parameters, evaluated at $\mathbf{p}$.
$\tilde{\mathbf{g}}(\mathbf{p})$	subgradient of the cost function with respect to the parameters, evaluated at $\mathbf{p}$.
$G(q, \mathbf{p})$	filter in the Box and Jenkins model.
$\mathcal{h}$	entropy.
$\mathbf{H}(\mathbf{p})$	Hessian of the cost function with respect to the parameters, evaluated at $\mathbf{p}$.
$\mathbf{H}(s, \mathbf{p})$	transfer matrix of a model with parameters $\mathbf{p}$.
$h(t)$	impulse response.
$H(t)$	Heaviside step $(H(t) = 0, t < 0; H(t) = 1, t \geq 0)$.
$\mathbf{H}_{\mathrm{a}}(\mathbf{p})$	approximate Hessian (for a quadratic cost), evaluated at $\mathbf{p}$.
i.i.d.	independently identically distributed.
$\mathbf{I}_n$	identity matrix $(n \times n)$.
j	imaginary number such that $j^2 = -1$.
j	cost function to be optimized.

$\mathbf{k}(t)$	correction gain for recursive least squares.
$\ker \mathbf{M}$	kernel of $\mathbf{M}$.
$\mathbf{K}_t$	gain of the Kalman filter.
$\mathcal{L}$	Lagrangian
LI	linear in its inputs.
ln	napierian logarithm.
LP	linear in its parameters.
$\mathbb{M}$	set of model structures.
m	design measure.
M	model structure.
$M(\mathbf{p})$	model with structure M and parameter vector $\mathbf{p}$.
MA	moving average.
MAP	maximum *a posteriori.*
$\mathrm{med}\{.\}$	median.
MSE(.)	mean-square error.
$M_1(\mathbf{p}_1) = M_2(\mathbf{p}_2)$	$M_1(\mathbf{p}_1)$ and $M_2(\mathbf{p}_2)$ have the same input-output behaviour.
$\mathbf{n}$, $n(t)$	noise or perturbations.
n_a	number of parameters in polynomial $A(q, \mathbf{p})$.
n_b	number of parameters in polynomial $B(q, \mathbf{p})$.
n_c	number of parameters in polynomial $C(q, \mathbf{p})$.
n_d	number of parameters in polynomial D(q, $\mathbf{p}$).
n_f	number of factors (or operating conditions).
n_l	number of the parameters on which the model output depends linearly.
n_nl	number of the parameters on which the model output depends nonlinearly.
n_p	number of parameters (dimension of $\mathbf{p}$).
n_r	input-output delay in a discrete time model.
n_sd	number of significant digits
n_t	number of measurements (or dimension of $\mathbf{y}^\mathrm{s}$).
n_u	number of inputs (dimension of $\mathbf{u}$).
n_x	number of state variables (dimension of $\mathbf{x}$).
n_y	number of outputs (dimension of $\mathbf{y}$).
n_ξ	dimension of $\boldsymbol{\xi}$.
$\mathcal{N}(m, v)$	distribution of a Gaussian random variable with mean m and variance v.
$\mathcal{N}(\mathbf{m}, \mathbf{V})$	distribution of a Gaussian random vector with mean $\mathbf{m}$ and covariance $\mathbf{V}$.
o(x)	term such that o(x)/x tends to zero with x.
$\mathbf{O}^\perp$	orthogonal projector.
$\mathbf{p}$	vector of parameters to be estimated (of dimension n_p).
$\mathbb{P}$ or $\mathbb{P}_0$	prior feasible set for $\mathbf{p}$.
$\mathbf{p}^*$	true value of $\mathbf{p}$.
$\hat{\mathbf{p}}$	estimate of $\mathbf{p}$ associated with a local optimum of the cost function.
$\check{\mathbf{p}}$	estimate of $\mathbf{p}$ associated with a global optimum of the cost function.
$[\mathbf{p}]$	axes-aligned box of $\mathbb{R}^{n_\mathrm{p}}$ (interval vector).

$\mathbf{p}_{\mathrm{e}}$	extended parameter vector.
$\hat{\mathbf{p}}_{\mathrm{iv}}$	instrumental variable estimator.
$\hat{\mathbf{p}}^{k}$	parameter estimate at iteration k.
$\mathbf{p}^{\mathrm{l}}$	vector of the parameters on which the model output depends linearly.
$\hat{\mathbf{p}}_{\mathrm{lm}}$	least-moduli estimator.
$\hat{\mathbf{p}}_{\mathrm{ls}}$	least-squares estimator.
$\hat{\mathbf{p}}_{\mathrm{m}}$	M-estimator.
$\hat{\mathbf{p}}_{\mathrm{map}}$	maximum *a posteriori* estimator.
$\hat{\mathbf{p}}_{\mathrm{ml}}$	maximum-likelihood estimator.
$\hat{\mathbf{p}}_{\mathrm{mm}}$	minimax estimator.
$\hat{\mathbf{p}}_{\mathrm{mr}}$	minimum-risk estimator.
$\mathbb{P}_{n}$	posterior feasible set for $\mathbf{p}$ with n data points.
$\mathbf{p}^{\mathrm{nl}}$	vector of the parameters on which the model output depends nonlinearly.
prob(X)	probability of event X.
$\mathbf{P}_{t}$ or $\mathbf{P}(t)$	covariance matrix for the state (Kalman filter) or parameters (recursive least squares).
P_{u}	average power of input signal u.
$p_{\mathrm{u}}(\omega)$	power spectral density of signal u.
PUI	parameter uncertainty interval.
$\mathbf{Q}$	weighting matrix (symmetric and non-negative-definite).
$\mathbb{Q}_{k}$	convex polyhedron associated with k linear inequalities.
q^{-1}	unit delay operator, such that $q^{-1}f(t) = f(t-1)$.
$q_{\Xi}(\hat{\mathbf{p}}_{\mathrm{ls}}, \mathbf{p}^{*})$	approximation of the distribution of $\hat{\mathbf{p}}_{\mathrm{ls}}$ for a true value $\mathbf{p}^{*}$ of the parameter vector and an experiment Ξ.
$\mathbf{r}$	regressor vector for an LP model.
$\mathbf{R}$	regressor matrix.
$\mathbb{R}$	set of real numbers.
$\mathbb{R}^{+}$	set of non-negative real numbers.
Re(x)	real part of x.
s	Laplace operator.
S	system.
s.d.	structurally distinguishable.
s.g.i.	structurally globally identifiable.
s.l.i.	structurally locally identifiable.
s.u.i.	structurally unidentifiable.
$\mathbf{s}_{\mathrm{e}}(t, \mathbf{p})$	sensitivity of error $e(t, \mathbf{p})$ with respect to $\mathbf{p}$.
$\mathbb{S}_{\mathrm{exp}}$	expectation surface (locus of model responses $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ when $\mathbf{p}$ varies).
sign a	function taking the value 1 if $a > 0$, 0 if $a = 0$ and -1 if $a < 0$.
$\mathbb{S}_{\mathrm{resp}}$	response surface $\{(\boldsymbol{\xi}^{\mathrm{T}}, y_{\mathrm{m}}(\boldsymbol{\xi}))^{\mathrm{T}}, \boldsymbol{\xi} \in \mathbb{\xi}\}$.
sup	supremum
supp {.}	set of all support points.
$\mathbf{s}_{\mathrm{y}}(t, \mathbf{p})$	sensitivity of model output $y_{\mathrm{m}}(t, \mathbf{p})$ with respect to $\mathbf{p}$.
t, t_{i}	independent variable (often time), which may take discrete values.

T	sampling period for a continuous-time system.
$^{\mathrm{T}}$	transposition operator, applies to the matrix or vector on its left.
trace $\mathbf{M}$	sum of the diagonal entries of square matrix $\mathbf{M}$.
$\mathbf{u}$	vector of (controlled) inputs of the system (of dimension n_{u}).
$\mathcal{U}(a, b)$	uniform distribution between a and b.
vol($\mathbb{S}$)	volume of set $\mathbb{S}$
w_i	weighting coefficient (≥ 0).
$\mathbf{x}$	state vector (of dimension n_{x}).
$\mathbf{x}_{\mathrm{e}}$	extended state vector.
$\mathbf{y}$	vector of system outputs.
$\mathbf{y}(t)$ or $\mathbf{y}_t$	vector of system outputs observed at t (of dimension n_{y}).
$\hat{y}(t\|t-1, \mathbf{p})$	prediction of system output at t, given the information available at $t-1$ for a value $\mathbf{p}$ of the model parameter vector.
$\mathbf{y}^{\mathrm{m}}(\mathbf{p})$	vector of model outputs corresponding to $\mathbf{y}^{\mathrm{s}}$ computed for the value $\mathbf{p}$ of the parameter vector.
$\mathbf{y}_{\mathrm{m}}(t, \mathbf{p})$	vector of model outputs corresponding to $\mathbf{y}(t)$.
$\mathbf{y}^{\mathrm{s}}$	vector of all outputs measured on the system since the initial time.
z	z-transform variable.
$\mathbf{z}(t)$	vector of the values at t of quantities of interest.
$\mathbf{z}_{\mathrm{m}}(t, \mathbf{p})$	estimate of $\mathbf{z}(t)$ produced by the model with parameters $\mathbf{p}$.
$\delta(.)$	Dirac delta.
$\delta_{i,j}$	Kronecker symbol ($\delta_{i,j} = 0$ if $i \neq j$; $\delta_{i,j} = 1$ if $i = j$).
Δx	variation of x.
$\{\varepsilon(t)\}$	sequence of independent random variables.
$\{\eta(t)\}$	sequence obtained by passing $\{\varepsilon(t)\}$ through a linear filter.
μ_i	weight of the ith support point ξ^i in a discrete design measure.
Ξ	experimental conditions for all measurements $\mathbf{y}^{\mathrm{s}}(\Xi)$.
$\mathbb{\Xi}$	feasible domain for Ξ.
ξ^i	experimental conditions for the ith measurement $\mathbf{y}(\xi^i)$ and ith support point of a discrete design measure.
$\mathbb{\xi}^i$	feasible domain for ξ^i.
$\mathbf{\Pi}$	projection matrix.
π	probability density.
$\pi(y, \mathbf{p})$	joint probability density of y and $\mathbf{p}$.
$\pi_{\mathrm{p}}(\mathbf{p})$	prior probability density of the parameters.
$\pi_{\mathrm{y}}(y\|\mathbf{p})$	probability density of y if the data originate from a model with parameters $\mathbf{p}$ (likelihood of y).
π_{ε}	probability density of ε.
σ	standard deviation (measurement noise).
$\mathbf{\Sigma}$	covariance matrix (measurement noise).
$\chi^2(n)$	chi-square distribution with n degrees of freedom.
$\chi^2_{\alpha}(n)$	numerical value with a probability α of being exceeded by a random variable distributed $\chi^2(n)$.
$\psi(\omega)$	spectral density of noise η.
ω	angular frequency.
$\mathbf{\Omega}$	prior covariance of parameter vector $\mathbf{p}$.

Convention for derivatives (Vetter, 1970, 1973)

$\dfrac{\partial \mathbf{A}}{\partial b}$ derivative of matrix $\mathbf{A}$ with respect to scalar b (matrix with the same dimensions as $\mathbf{A}$, the (i, j) entry of which is $\partial a_{i,j}/\partial b$).

$\dfrac{\partial \mathbf{A}}{\partial \mathbf{B}}$ derivative of matrix $\mathbf{A}$ (possibly a vector or a scalar) with respect to matrix $\mathbf{B}$ (possibly a vector or a scalar). If $\dim \mathbf{A} = n_\mathrm{A} \times m_\mathrm{A}$ and $\dim \mathbf{B} = n_\mathrm{B} \times m_\mathrm{B}$, then $\partial \mathbf{A}/\partial \mathbf{B}$ is an $n_\mathrm{A}n_\mathrm{B} \times m_\mathrm{A}m_\mathrm{B}$ matrix, obtained by putting $\partial \mathbf{A}/\partial b_{i,j}$ in (i, j) position. Some important examples follow.

$\dfrac{\partial j}{\partial \mathbf{p}} = \mathbf{g}$ gradient of the cost function with respect to the parameters (column vector with n_p entries).

$\dfrac{\partial j}{\partial \mathbf{p}^\mathrm{T}} = \mathbf{g}^\mathrm{T}$ transposed of the gradient of the cost function with respect to the parameters (row vector with n_p entries).

$\dfrac{\partial^2 j}{\partial \mathbf{p} \partial \mathbf{p}^\mathrm{T}} = \mathbf{H}$ Hessian of the cost function ($n_\mathrm{p} \times n_\mathrm{p}$).

1 Introduction

1.1 Aims of modelling

Model and modelling are catchwords with many different interpretations. In oncology, for instance, a model of a cancer is an animal in which this cancer can be triggered. In this book, a *model* will be a *mathematical description of a real process*, built with a definite aim in mind. This aim may be:

- analysing phenomena to deepen understanding of them (models in physics, chemistry...);
- estimating quantities for which no sensor is available, from indirect measurements;
- testing hypotheses (medical or fault diagnosis, on-line quality control...);
- teaching (simulators for aircrafts, nuclear power plants, patients in critical condition...);
- predicting short-term behaviour (adaptive control of time-varying processes) or long-term behaviour (economic forecasting for governmental planning);
- controlling processes (regulation around some nominal set-point, trajectory following with large transients, optimal control...);
- processing signals (noise cancellation, data compression, filtering, interpolation...). The implementation of a Kalman filter, for instance, requires a model of the process generating the data.

Whatever this aim may be, it should always be made explicit, because it should very strongly influence the modelling procedure. Models used to tune the coefficients of a PID controller, for instance, are quite different from those employed to study chemical reactions in detail. As a result, the problems raised by the building of these models will have little in common. Since modelling is most often an interdisciplinary activity, it is of paramount importance that the aim(s) of the exercise be clear for all those involved. Ultimately, the model obtained should be judged according to whether these aims have been satisfactorily attained. It may happen that model building, by the questions that it raises, allows one to solve problems that had not been formulated at the start. Of course, this should not serve as an excuse for not precisely stating the objectives to be achieved.

1.2 System

A *system* (or *process*) is a part of the universe, which we have chosen, more or less arbitrarily, to consider as an entity with which we interact (Figure 1.1).

— We *observe* some characteristic quantities of the system, and the results of these observations form the *output* vector $\mathbf{y}$. These quantities may depend on a vector $\boldsymbol{\xi}$ of independent variables, which often reduces to the time t at which the measurements are made. What will then be available is the value of $\mathbf{y}$ for specific (and known) values of $\boldsymbol{\xi}$, which we shall indifferently denote by $\mathbf{y}_{\boldsymbol{\xi}}$ or $\mathbf{y}(\boldsymbol{\xi})$. In what follows, $\boldsymbol{\xi}$ will often be replaced by t when it corresponds to a single independent variable, even if this variable has nothing to do with time.

— We *are interested in* some characteristic quantities of the system, which we shall denote by $\mathbf{z}$ and which may depend on $\boldsymbol{\xi}$ or t. The vector $\mathbf{z}$ may include some unmeasurable quantities, and thus differ from $\mathbf{y}$.

— We *act on* the system by means of some quantities, the evolution of which is known and more or less under our control. These quantities are the *inputs*, which we shall denote by $\mathbf{u}$.

— We *endure* the action on the system (or on the measurements taken from the system) of quantities that are not under control and are more or less unknown. These quantities are the *perturbations*, or *noises*, which we shall denote by $\mathbf{n}$.

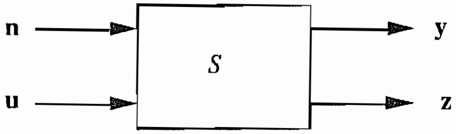


Figure 1.1. System S

Consider for instance a system consisting of a chemical reactor within which a mixture of gases is circulated over a solid catalyst. The components of the output vector $\mathbf{y}$ may be the partial pressures of various gases at the outlet, the pressure inside the reactor and the temperature at various locations. The vector $\mathbf{z}$ of the quantities of interest may include surface concentrations on the catalyst that cannot be measured directly. The components of the input vector $\mathbf{u}$ may be the controls of the heating, of the pump used to circulate the gaseous mixture and the partial pressures of the various gases at the inlet. Perturbations may correspond to uncertainties about the value of $\mathbf{u}$ and $\mathbf{y}$, to catalyst poisoning, and so on.

Deciding what will be considered as a system is often far from obvious. After Descartes, we have become used to the idea of splitting a problem into as many subproblems as necessary to make them tractable. Unfortunately, such a procedure often does not apply. In biology, for instance, it is impossible to isolate part of a living organism without modifying its behaviour. Similarly, there are unstable systems which cannot be studied safely in open loop. Moreover, it is often difficult, in a complex system with feedback, to distinguish causes (inputs) from effects (outputs). One may then pool all measurable signals into a single vector without classifying them *a priori* as input or output (Willems, 1986a; 1986b; 1987). Throughout this book, however, we shall question neither the definition of the system to be studied nor the classification of the measured signals as inputs or outputs (with the exception of Section 4.1.7).

1.3 Model

The model M of the system is a rule to compute, from quantities known *a priori* or measured from the system, other quantities that we are interested in and which we hope will resemble their actual values in the system. Frequently, the model computes, from the input $\mathbf{u}$ of the system, an output $\mathbf{y}_m$ which should resemble $\mathbf{y}$ as closely as possible. If $\mathbf{z}$ differs from $\mathbf{y}$, the model may also compute a vector $\mathbf{z}_m$ which may, under identifiability conditions to be considered later, resemble $\mathbf{z}$. Since the model and system have the same input, the model is said to be *parallel* (Figure 1.2).

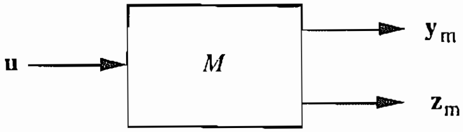


Figure 1.2. Parallel model

Less traditional configurations can also be considered, such as a model which, from the output $\mathbf{y}$ of the system, computes a vector $\mathbf{u}_m$ which should resemble the input $\mathbf{u}$ as closely as possible. This is a *series* or *inverse model* (Figure 1.3).

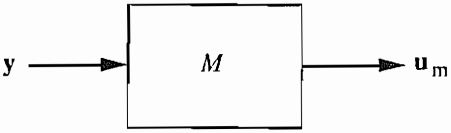


Figure 1.3. Series or inverse model

Models combining series and parallel parts can also be constructed. Whatever the structure chosen for the model, it will in general involve unknown quantities, usually assumed to be constant but sometimes liable to vary, to be estimated from available prior knowledge and data. These quantities are the *parameters* $\mathbf{p}$. One then speaks of a *parametric model*, the main type of model to be considered in what follows. We shall distinguish the model *structure* M from the specific model $M(\mathbf{p})$ obtained by setting its parameters to some specific numerical value $\mathbf{p}$.

The choice of M, also called *characterization*, is a critical step in modelling. Chapter 2 will give some indications of the choices to be made at this stage and the tools that can be used to study properties of model structures.

Once the model structure has been selected, its parameters must be chosen according to a specified criterion, usually the optimization of some cost function. If several model structures compete for the description of the same data, their performance will also be compared with the help of a criterion.

1.4 Criterion

Suppose, to fix ideas, that the model is of parallel type, *i.e.* subjected to the same inputs and initial conditions as the system, if the latter are known. (Otherwise, the unknown initial conditions will be incorporated into the parameter vector **p**, or taken as zero if the system is stable enough for their transient effect to be neglected.) The difference between the system and model outputs

$$\mathbf{e}_y(t, \mathbf{p}) = \mathbf{y}(t) - \mathbf{y}_m(t, \mathbf{p})$$

is then called *output error* (Figure 1.4).

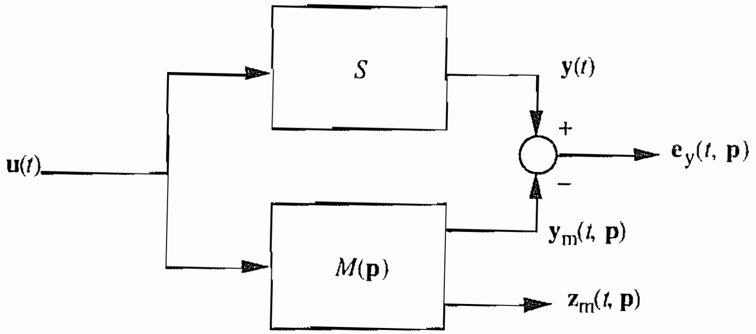


Figure 1.4. Output error for a parallel model

Most often, one wishes this output error to be as close to **0** as possible, which raises the difficult question of the definition of the scale of values to compare the performance of competing models. This scale will take the form of a scalar function j of the parameters and possibly of the structure, called the *cost function*. Assume that the cost is to be minimized. (If a cost function is to be maximized, changing its sign transforms it into a cost to be minimized.) $M_1(\mathbf{p}_1)$ is then better than $M_2(\mathbf{p}_2)$ *in the sense of the criterion associated with j* if

$$j(M_1(\mathbf{p}_1)) < j(M_2(\mathbf{p}_2)).$$

The choice of the criterion should reflect why the model is built. In fact, this purpose will then be momentarily forgotten and replaced by another one, easier to achieve, namely optimization of the cost function j with respect to the parameters (and possibly the model structure). Chapter 3 will be devoted to various types of criteria and their properties.

Once the cost function has been chosen, the next step is its optimization.

1.5 Optimization

The *optimization algorithm* uses the available information to evaluate the best value $\hat{\mathbf{p}}$ of the parameters (and possibly the best model structure $\hat{M}$) in the sense of minimizing the cost j. The flow of information may, for instance, be as indicated in Figure 1.5.

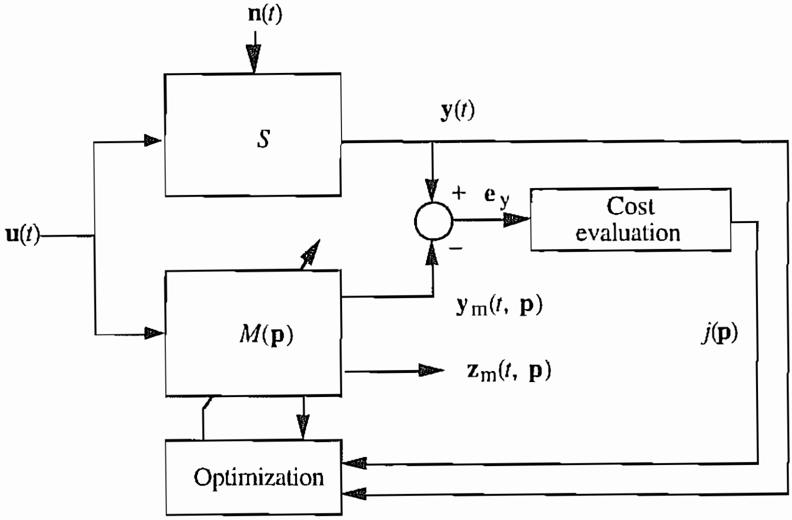


Figure 1.5. Possible flow of information for optimization

When the cost is quadratic in an error that is affine in the parameters, explicit formulas can be derived for $\hat{\mathbf{p}}$, in the celebrated least-squares method. When such explicit formulas are unavailable, optimization is usually performed iteratively. Starting from $\hat{\mathbf{p}}^k$, the estimate of $\hat{\mathbf{p}}$ at iteration k, which makes the value of the cost $j(\hat{\mathbf{p}}^k)$, the algorithm computes $\hat{\mathbf{p}}^{k+1}$ such that $j(\hat{\mathbf{p}}^{k+1}) < j(\hat{\mathbf{p}}^k)$. This raises two critical issues, *initialization* (choice of $\hat{\mathbf{p}}^0$) and *termination* of the iterating process. Sometimes, the data are to be processed successively (possibly in real time). One then speaks of *on-line* algorithms, as opposed to *off-line* algorithms, where all data are processed in a single batch. *Local* algorithms, which may get trapped in some neighbourhood of $\hat{\mathbf{p}}^0$ and thus only reach a local optimum of the cost, should be distinguished from *global* algorithms, which aim to find arguments corresponding to the global optimum of the cost over the prior feasible domain for the parameters.

A selection of algorithms will be presented in Chapter 4.

1.6 Parameter uncertainty

It would be naive to consider that $\hat{\mathbf{p}}$ resulting from the optimization procedure corresponds to the only model worthy of consideration. Even assuming that the structure of the process is M, which is never exactly true, there is a *set of acceptable* *models*, given the uncertainties in the measurements. Chapter 5 will present various techniques to characterize this set.

The set of all acceptable models depends on the experiments performed to collect the data, *i.e.* on the choices made as regards the input shape, location and type of sensors and actuators, measurement schedule, data processing, and so on. One should therefore *design experiments* so as to collect the most pertinent information. Some tools for this purpose will be presented in Chapter 6.

1.7 Critical analysis of the results

Finally, some critical analysis of the results obtained is essential. Modelling abounds with provisional choices about the experimental set-up, the model structure, the cost function, the optimization algorithm, and so on, which should be challenged. The model should therefore be submitted to tests attempting to *invalidate* (or *falsify*) it. If serious defects of the model are revealed, the choices made so far should be questioned. Chapter 7 will describe various techniques to prove that a model is inadequate. Unfortunately, none exists to prove that a model is the best that could be obtained!

1.8 In summary

Building a parametric model from experimental data consists of six basic steps:

— collecting data (Chapter 6),
— choosing the model structure(s) (Chapter 2),
— defining a quality criterion (Chapter 3),
— optimizing the associated cost function to get an optimal numerical value for the parameters, and possibly select the most suitable model structure (Chapter 4),
— evaluating the uncertainty in the estimated parameters (Chapter 5),
— questioning the results (Chapter 7).

One should not deduce from this enumeration that these steps are performed successively and in this order. Nothing, for instance, requires that the data be collected before some reflection takes place about the type of model to be employed. Similarly, critical analysis of the results should constantly be in the mind of the model builder. Realizing the arbitrary nature of some of our choices, we should be ready to modify them when confrontation with reality demonstrates their inadequacy.

2 Structures

The choice of a structure for the mathematical model is called *model-structure selection*, or more concisely *characterization*. One might, for instance, choose the structure described by the first-order linear differential equation:

$$\frac{\mathrm{d}y_\mathrm{m}}{\mathrm{d}t} = -p_1 y_\mathrm{m} + p_2 u, \quad y_\mathrm{m}(0) = 0,$$

with positive p_1 and p_2. One thus defines a class of possible behaviour and a prior feasible set to which the parameter vector $\mathbf{p}$ must belong for the model to be considered acceptable. Let M and $M(\mathbf{p})$ respectively denote the model structure and the model with structure M and parameters $\mathbf{p}$. The prior feasible set for $\mathbf{p}$ will be denoted by $\mathbb{P}$. In what follows, $\mathbb{P}$ will usually be either $\mathbb{R}^{n_\mathrm{p}}$ or a subset of $\mathbb{R}^{n_\mathrm{p}}$ defined by a finite set of inequality constraints.

Characterization is critical, because intuition plays an important part. It is of paramount importance, since it defines the choice available for the selection of the "best model". Only models with a finite number of parameters will be considered, described by algebraic, differential or finite-difference equations.

Various classes of models will be distinguished in this chapter. It will be necessary to choose among these classes, taking into account

— the aim of the modelling (Chapter 1),
— the conditions under which the model is going to be employed (operating ranges, nature of inputs, communication with other elements of a control system...),
— the cost of building the model,
— the information available (there is no point in conceiving a very complex model with many parameters if data are scarce and imprecise).

REMARK 2.1

Most models to be considered will implicitly assume that if the system initially at rest receives an input $\mathbf{u} \equiv \mathbf{0}$, its output $\mathbf{y}_\mathrm{m}$ will be zero in the absence of any perturbation. Inputs and outputs must then be expressed in a coordinate system that satisfies this assumption (using deviations from equilibrium conditions). ◊

2.1 Phenomenological and behavioural models

This distinction is rather schematic, but brings out two types of modelling that translate into differing requirements.

Phenomenological (or *knowledge-based*) *models* are familiar to anyone who has attended physics or chemistry courses; see, *e.g.*, the survey paper by Box and Hunter (1965). They are built from basic principles by writing down conservation or balance equations (for mass, momentum, energy...). The fact that this type of model is built from physical considerations facilitates incorporation of prior information and *a posteriori* checking of the orders of magnitude of the estimated parameters. Consider, for instance, the chemical reaction described by Figure 2.1.

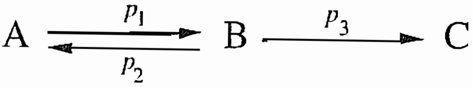


Figure 2.1. Chemical reaction

Under the hypotheses that all elementary reactions obey first-order kinetics and that the reactor is isothermal and well stirred, it can be associated with the set of equations

$$\frac{\mathrm{d}[A]}{\mathrm{d}t} = -p_1[A] + p_2[B],$$
$$\frac{\mathrm{d}[B]}{\mathrm{d}t} = p_1[A] - (p_2 + p_3)[B],$$
$$\frac{\mathrm{d}[C]}{\mathrm{d}t} = p_3[B].$$

The model structure is thus imposed by the prior knowledge (or hypotheses) about the system studied, which does, as we shall see, sometimes raise specific problems. The parameters p_i are the kinetic constants of the elementary reactions, and the state variables $[A]$, $[B]$ and $[C]$ are the concentrations of the reacting entities. All of them therefore have a precise concrete meaning. As soon as the process to be studied becomes somewhat complex, the model state may be of very high dimension (resulting for instance from the discretization of partial differential equations). The model may thus consist of many equations, often nonlinear. The simulation of such models generally takes a lot of time on powerful computers, and they are therefore seldom used directly to compute a control law. On the other hand, they are well suited to detailed simulation for the prediction of long-term behaviour or for gaining further insight into the internal working of the process. Some computer codes for the simulation of nuclear power plants, for example, are so complex that they cannot be run in real time for the training of operators. One must then resort to simplified models.

At the other end of the spectrum, one finds *behavioural models*, which merely approximate observed behaviour without requiring any prior knowledge of the process that generated the data. It is not even necessary to know what the inputs and outputs stand for or in what units they are expressed. The model structure does not claim to correspond in any more fundamental way to that of the process, and the parameters have no physical meaning. If, for example, an experimental curve is described by the polynomial

$$y_{\mathrm{m}}(t, \mathbf{p}) = p_1 + p_2 t + p_3 t^2 + p_4 t^3 + \dots$$

it is possible to reproduce with arbitrary precision any finite set of experimental data $y(t_i)$, $i = 1, \dots, n_t$, provided that the degree of the polynomial is large enough. This is a particularly simplistic example of a behavioural model, with very poor predictive capability. There are, of course, more sophisticated methods of building models, the aim of which is still to reproduce input-output behaviour independently of any knowledge of the underlying process (Sections 2.4 and 3.3.5). One class deserving particular mention is that of methods aimed at building a state-space representation associated with a given input-output behaviour (so-called *realization methods*). Realization methods are available for linear systems (Ho and Kalman, 1966; Dang Van Mien, 1973; Van Overschee and De Moor, 1994, 1996; De Moor and Van Overschee, 1995) and for some classes of nonlinear systems, such as bilinear systems (Fliess, 1978). For nonlinear systems which can be approximated by linear models around operating points characterized by a measurable mode, a unique model with a state-affine structure can be built from a family of linear realizations associated with a set of operating points (Dang Van Mien and Normand-Cyrot, 1984). Note that the hope of having some day at one's disposal a general method applicable to any nonlinear system is vain, as demonstrated by the existence of universal differential equations (Rubel, 1981), capable of approaching with arbitrary precision any continuous behaviour with a model that depends on five parameters only. The sensitivity of such a solution to variation of these parameters is of course extreme, and the model obtained has no predictive power. Neural networks are also capable, in principle, of approximating any continuous behaviour with arbitrary precision (Hornik, Stinchcombe and White, 1989), and one should be cautious as regards their predictive abilities for the same reasons.

Behavioural models are in general simpler to simulate and more suited to the computation of controls than phenomenological models. Table 2.1 summarizes the usual properties of these two types of model (although exceptions can easily be found).

	Phenomenological models	Behavioural models
Parameters	have a concrete meaning	have no concrete meaning
Simulation	long and difficult	quick and easy
Prior information	taken into account	neglected
Validity domain	large (if structure is correct!)	restricted

Table 2.1. Phenomenological and behavioural models

The choice between phenomenological and behavioural models is not always as simple as this table might suggest. Estimating the few parameters of a phenomenological model (for instance a suitably discretized partial differential equation) may turn out to be simpler than estimating the many parameters of a multi-input multi-output behavioural model which would not exploit the physical laws governing the process studied.

2.2 Linear and nonlinear models

Two types of linearity must be distinguished. Let $\mathbf{y}_{\mathrm{m}}(t, \mathbf{p}, \mathbf{u})$ be the output at time t of the model with parameters $\mathbf{p}$ when the input $\mathbf{u}(\tau)$, $0 \leq \tau \leq t$, has been applied from a zero initial condition. A model structure will be said to be *linear in its inputs* (*LI*) if its outputs satisfy the superposition principle with respect to its inputs, *i.e.* if

$$\forall\ (\lambda, \mu) \in \mathbb{R}^2,\ \forall\ t \in \mathbb{R}^+,\ \mathbf{y}_{\mathrm{m}}(t, \mathbf{p}, \lambda\mathbf{u}_1 + \mu\mathbf{u}_2) = \lambda\mathbf{y}_{\mathrm{m}}(t, \mathbf{p}, \mathbf{u}_1) + \mu\mathbf{y}_{\mathrm{m}}(t, \mathbf{p}, \mathbf{u}_2).$$

When control engineers speak of linear models, they usually refer to this type of linearity. Moreover, they often assume implicitly that the model is time-invariant, *i.e.* that its behaviour is invariant under a translation of the origin of time.

A model structure will be said to be *linear in its parameters* (*LP*) if its outputs satisfy the superposition principle with respect to its parameters, *i.e.* if

$$\forall\ (\lambda, \mu) \in \mathbb{R}^2,\ \forall\ t \in \mathbb{R}^+,\ \mathbf{y}_{\mathrm{m}}(t, \lambda\mathbf{p}_1 + \mu\mathbf{p}_2, \mathbf{u}) = \lambda\mathbf{y}_{\mathrm{m}}(t, \mathbf{p}_1, \mathbf{u}) + \mu\mathbf{y}_{\mathrm{m}}(t, \mathbf{p}_2, \mathbf{u}).$$

When statisticians speak of linear models, they usually refer to this type of linearity; see, *e.g.*, the survey by Jennrich and Ralston (1979).

A model structure will be said to be *affine in its parameters* (*AP*) if its output satisfies

$$\mathbf{y}_{\mathrm{m}}(t, \mathbf{p}, \mathbf{u}) = \mathbf{y}_{\mathrm{m}_1}(t, \mathbf{u}) + \mathbf{y}_{\mathrm{m}_2}(t, \mathbf{p}, \mathbf{u}),$$

where $\mathbf{y}_{\mathrm{m}_2}(t, \mathbf{p}, \mathbf{u})$ is LP. AP structures only differ from LP structures by the addition of a term independent of the parameters, so the methods available for the study of LP structures extend without difficulty to AP structures.

It is useful to know whether the structure considered is LP or not, and LI or not, because this will have important consequences on the algorithms to be used.

EXAMPLE 2.1

$y_{\mathrm{m}}(t+1, p) = pu(t)$	is LP and LI,
$y_{\mathrm{m}}(t+1, p) = py_{\mathrm{m}}(t, p) + u(t)$	is non-LP and LI,
$y_{\mathrm{m}}(t+1, p) = pu^2(t)$	is LP and non-LI,
$y_{\mathrm{m}}(t+1, p) = py_{\mathrm{m}}^2(t, p) + u(t)$	is non-LP and non-LI,
$y_{\mathrm{m}}(t+1, p) = py(t) + u(t)$	is AP. ◊

Whenever possible, LP and LI structures will be preferred. LI structures benefit from the existence of very powerful mathematical results that facilitate their theoretical study (stability conditions, optimal control, effect of perturbations...). Estimating the parameters of LP structures is easy, and it is often possible to use explicit formulas that avoid any iterative procedure (Section 4.1). The evaluation of the uncertainty in the parameters and the design of experiments are also simplified (Chapters 5 and 6).

On the other hand, LI models often have a limited domain of validity, for most real processes become nonlinear when the amplitude of the inputs gets large enough. In such cases, an LI model may only approximate the behaviour of the system correctly around some operating point. As regards LP structures, their parameters often have no concrete meaning.

It is sometimes possible to transform a non-LP structure into an LP one by a change of variables (Box and Cox, 1964; Atkinson, 1985, 1995). Thus, for instance, the non-LP structure described by $y_{\mathrm{m}}(t_k, \mathbf{p}) = p_1 \exp(-p_2 t_k)$ becomes LP if written as

$$\ln y_{\mathrm{m}}(t_k, \mathbf{q}) = q_1 - q_2 t_k, \quad \text{where} \quad q_1 = \ln p_1 \quad \text{and} \quad q_2 = p_2.$$

Such a procedure, however, is not without consequences for the value of the estimates obtained. In the presence of noise, one will not get the same results estimating **p** directly or *via* the estimation of **q**. This point will be considered again in Example 4.6.

2.3 Continuous- and discrete-time models

2.3.1 Continuous-time models

The processes studied are generally assumed to evolve in a continuous time. Hence the traditional tendency is to employ models described by differential equations, and especially *differential state-space models* such as

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}, \mathbf{p}, \mathbf{u}, t), \quad \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}),$$
$$\mathbf{y}_\mathrm{m}(t) = \mathbf{h}(\mathbf{x}, \mathbf{p}, \mathbf{u}, t),$$

which in the time-invariant LI case becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{A}(\mathbf{p})\mathbf{x}(t) + \mathbf{B}(\mathbf{p})\mathbf{u}(t)\ , \quad \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}),$$
$$\mathbf{y}_\mathrm{m}(t) = \mathbf{C}(\mathbf{p})\mathbf{x}(t) + \mathbf{D}(\mathbf{p})\mathbf{u}(t).$$

One may pass from this last type of representation to a *transfer matrix* representation

$$\mathbf{y}_\mathrm{m}(s, \mathbf{p}) = \mathbf{H}_1(s, \mathbf{p})\mathbf{u}(s) + \mathbf{H}_2(s, \mathbf{p})\mathbf{x}_0(\mathbf{p}),$$

with

$$\mathbf{H}_1(s, \mathbf{p}) = \mathbf{C}(\mathbf{p})[s\mathbf{I} - \mathbf{A}(\mathbf{p})]^{-1}\mathbf{B}(\mathbf{p}) + \mathbf{D}(\mathbf{p})$$

and

$$\mathbf{H}_2(s,\mathbf{p}) = \mathbf{C}(\mathbf{p})[s\mathbf{I} - \mathbf{A}(\mathbf{p})]^{-1},$$

where s is the *Laplace operator* and **I** the identity matrix. Finally, by returning to the time domain, one gets a set of *input-output differential equations*

$$\sum_{i=0}^{n} \mathbf{P}_i(\mathbf{p}) \frac{\mathrm{d}^i}{\mathrm{d}t^i} \mathbf{y}_\mathrm{m} = \sum_{i=0}^{m} \mathbf{Q}_i(\mathbf{p}) \frac{\mathrm{d}^i}{\mathrm{d}t^i} \mathbf{u}.$$

For LI models, it is thus easy to pass from one type of representation to another. For non-LI models, even passing from a state-space representation to an input-output differential equation might become impossible.

REMARK 2.2

In addition to the usual state equations, one may have to consider algebraic constraints between state variables. Such will be the case, for instance, in chemical kinetics, when some reaction steps are so fast compared to others that they can be considered as at equilibrium. The resulting algebraic-differential set of equations can then be written as

$$\mathbf{M}(\mathbf{x}, \mathbf{p})\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}, \mathbf{p}, \mathbf{u}, t), \;\; \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p})$$

Usual state-space representations correspond to $\mathbf{M} = \mathbf{I}$, and purely algebraic systems to $\mathbf{M} = \mathbf{0}$. Numerical methods are available to solve such algebraic-differential systems for $\mathbf{x}$ (Hindmarsh, 1980; Bilardello *et al.*, 1993). ◊

2.3.2 Discrete-time models

The ever-increasing availability of computers has in many domains dealt a fatal blow to the supremacy of continuous-time models. The numerical simulation of discrete-time models is much simpler and quicker, which makes them well suited to real-time process control. Their use, however, may entail some loss of information on the behaviour of the underlying continuous-time system.

As in the continuous-time case, one may employ a *discrete-time state-space model*

$$\begin{aligned}&\mathbf{x}(t+1) = \mathbf{f}[\mathbf{x}(t), \mathbf{p}, \mathbf{u}(t), t], \;\; \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}),\\ &\mathbf{y}_\mathrm{m}(t) = \mathbf{h}[\mathbf{x}(t), \mathbf{p}, \mathbf{u}(t), t],\end{aligned}$$

where t is now an integer time index, which corresponds to actual time tT if the underlying continuous-time system is sampled with period T. When this model is LI and time-invariant, it can be written as

$$\begin{aligned}&\mathbf{x}(t+1) = \mathbf{A}(\mathbf{p})\mathbf{x}(t) + \mathbf{B}(\mathbf{p})\mathbf{u}(t), \;\; \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}),\\ &\mathbf{y}_\mathrm{m}(t) = \mathbf{C}(\mathbf{p})\mathbf{x}(t) + \mathbf{D}(\mathbf{p})\mathbf{u}(t).\end{aligned}$$

From this last type of representation, one may pass to a *transfer-matrix* representation

$$\mathbf{y}_\mathrm{m}(z, \mathbf{p}) = \mathbf{H}_1(z, \mathbf{p})\mathbf{u}(z) + \mathbf{H}_2(z, \mathbf{p})\mathbf{x}_0(\mathbf{p}),$$

with

$$\mathbf{H}_1(z, \mathbf{p}) = \mathbf{C}(\mathbf{p})[z\mathbf{I} - \mathbf{A}(\mathbf{p})]^{-1}\mathbf{B}(\mathbf{p}) + \mathbf{D}(\mathbf{p})$$

and

$$\mathbf{H}_2(z, \mathbf{p}) = \mathbf{C}(\mathbf{p})[z\mathbf{I} - \mathbf{A}(\mathbf{p})]^{-1}z,$$

where $z = \exp(Ts)$ is the Laplace transform of the forward-shift operator. This transfer-matrix representation easily translates into a *recurrence equation*

$$\mathbf{y}_\mathrm{m}(t+1) = -\sum_{i=0}^{n-1} \mathbf{P}_i(\mathbf{p})\mathbf{y}_\mathrm{m}(t-i) + \sum_{k=0}^{m-1} \mathbf{Q}_k(\mathbf{p})\mathbf{u}(t-k).$$

To write recurrence equations in a more condensed way, it is convenient to use the *delay operator* q^{-1}, such that $q^{-1}\mathbf{x}(t) = \mathbf{x}(t-1)$. This operator could also have been denoted by z^{-1}, but the usual notation q^{-1} emphasizes that what is meant is merely a condensed notation for a time-domain equation. Thus, for instance, the recurrence equation

$$y_\mathrm{m}(t+1) = -a_1 y_\mathrm{m}(t) - a_2 y_\mathrm{m}(t-1) + b_1 u(t) + b_2 u(t-1),$$

can also be written as

$$A(q^{-1})y_m(t+1) = B(q^{-1})u(t+1),$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} \text{ and } B(q^{-1}) = b_1q^{-1} + b_2q^{-2}.$$

Since discrete-time models are much easier to simulate numerically than continuous-time ones, it is possible to push the experimental study of their properties much further. On the other hand, they impose constraints on the measurement times, which must correspond to discrete times at which the model output is computed. Moreover, they may hide oscillations of the associated continuous-time system. Finally, their parameters generally do not have any clear physical meaning. In particular, the values of the parameters of a model obtained by discretizing a continuous-time model depend on the sampling period chosen.

The properties of discrete- and continuous-time models are not always analogous. One can, for example, create oscillating first-order LI discrete-time models, such as $y_m(t+1) = -p_1y_m(t) + u(t)$, with $0 < p_1 \leq 1$, whereas first-order LI differential equations do not oscillate. As a consequence, building a continuous-time model through the construction of an intermediary discrete-time model may turn out to be difficult. One may indeed get a discrete-time model with no continuous-time counterpart, and specific methods for continuous-time systems are still of interest (Unbehauen and Rao, 1987). Table 2.2 summarizes the properties of these two types of models.

	Continuous time	Discrete time
Parameters	independent of measurement times	depend on sampling period
Computer simulation	requires discretization	easy
Prior information	can be incorporated readily	mostly not taken into account (steady-state gain an exception)
Measurement times	individually chosen	dictated by choice of sampling period

Table 2.2. Continuous- and discrete-time models

2.3.3 Sampling

When a discrete-time model is built from data collected from a continuous-time process every T seconds, it is essential to make sure that the signal thus sampled satisfies the *Shannon condition* (no components at frequencies higher than $1/(2T)$). Otherwise, the *frequency folding* effect (or *aliasing*) may render the data useless, because all components of the continuous signal at frequencies higher than $1/(2T)$ will get superimposed on the useful frequency band. When in doubt as to the frequency content of the continuous signal to be sampled, one should insert a continuous-time (analogue) low-pass filter before sampling, in order to eliminate all components of the signal at frequencies higher than $1/(2T)$. This point will be considered again in Section 6.3.3.

To describe accurately the evolution of a continuous-time process with a discrete-time model, one may be led to adopt a small sampling period T compared to the time constants of the model. This may raise critical numerical difficulties when the model is implemented on a computer. Consider, for example, a single-input single-output LI continuous-time model. The poles and zeros of the transfer function associated with its discrete-time counterpart tend to cluster around the point $(1 + j0)$ when T tends to zero, because consecutive input and output samples get more and more similar. On a finite-precision computer, this results in a more and more marked deterioration of the quality of the simulation. The use of the operator $\delta = (q - 1)/T$ such that

$$\delta \mathrm{x}(t) = \frac{\mathrm{x}(t+1) - \mathrm{x}(t)}{T},$$

where the instant indexed by t for the discrete-time model corresponds to the instant tT for the continuous-time model, makes it possible to avoid these problems (Middleton and Goodwin, 1990). This operator approximates a first-order derivative by a finite difference. A discrete-time model using the operator δ will therefore tend to its continuous-time counterpart when T tends to zero. If T is small, this will ensure relative independence of the parameters of the discrete-time model with respect to T. A possible way to obtain a model in δ is:

— build a classical recurrence equation,
— write this model in condensed form using the q^{-1} operator,
— replace q by $1 + \delta T$,
— express the ratio of the output to the input as a δ transfer function,
— deduce a computing scheme involving elementary blocks δ^{-1}, the input $e(t+1)$ and output $s(t+1)$ of which satisfy

$$s(t+1) = \delta^{-1} e(t+1) = \frac{T}{q - 1}\, e(t+1) = \frac{Tq^{-1}}{1 - q^{-1}}\, e(t+1),$$

so $s(t+1) = s(t) + Te(t)$. The δ^{-1} operator is therefore a discrete-time integrator.

EXAMPLE 2.2

Let us apply this procedure to the recurrence equation

$$y_{\mathrm{m}}(t+1) = -a_1 y_{\mathrm{m}}(t) - a_2 y_{\mathrm{m}}(t-1) + b_1 u(t) + b_2 u(t-1).$$

We get

$$(q^2 + a_1 q + a_2) y_{\mathrm{m}}(t+1) = (b_1 q + b_2) u(t+1),$$
$$[(1 + 2\delta T + \delta^2 T^2) + a_1(1 + \delta T) + a_2] y_{\mathrm{m}}(t+1) = [b_1(1 + \delta T) + b_2] u(t+1),$$
$$[T^2 \delta^2 + (2T + a_1 T)\delta + (1 + a_1 + a_2)] y_{\mathrm{m}}(t+1) = [(b_1 T)\delta + (b_1 + b_2)] u(t+1),$$

which can also be written as the second-order δ transfer function

$$\frac{y_{\mathrm{m}}(t+1)}{u(t+1)} = \frac{\beta_1 \delta + \beta_2}{\delta^2 + \alpha_1 \delta + \alpha_2}.$$

The recurrence equation associated with this transfer function can be implemented with the scheme derived from Figure 2.2.

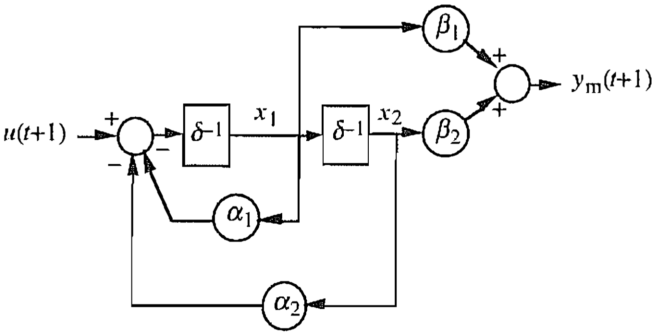


Figure 2.2. Discrete simulation scheme for a second-order δ transfer function

This scheme translates into

$$x_1(t+1) = \delta^{-1}[-\alpha_1 x_1(t+1) - \alpha_2 x_2(t+1) + u(t+1)],\ x_2(t+1) = \delta^{-1}[x_1(t+1)],$$
$$y_m(t+1) = \beta_1 x_1(t+1) + \beta_2 x_2(t+1),$$

i.e.

$$x_1(t+1) = x_1(t) + T[-\alpha_1 x_1(t) - \alpha_2 x_2(t) + u(t)],\ x_2(t+1) = x_2(t) + Tx_1(t),$$
$$y_m(t+1) = \beta_1 x_1(t+1) + \beta_2 x_2(t+1). \qquad \lozenge$$

This procedure extends without difficulty to higher orders. Numerically more robust results are however obtained by decomposing the δ transfer function into a product (or a sum) of first- and second-order terms and connecting the corresponding state-space submodels in series (or in parallel).

2.4 Deterministic and stochastic models

The models described in Section 2.3 are deterministic and describe the outputs as if they were uniquely determined by the inputs. This is often unrealistic, because of the various *perturbations* that act on the system or corrupt measurements. It is then necessary to find ways of describing the influence of these perturbations. A statistical description is usually used, where perturbations are described as stochastic processes (sequences of random variables). To simplify, we shall only consider systems with one deterministic input $u(t)$, one output $y(t)$ and one perturbation $n(t)$, described by recurrence equations. Depending on how the perturbation is assumed to act, many model structures can be obtained. We shall only mention the most commonly used (for more details, see, *e.g.*, (Ljung, 1987)). Assume that the data satisfy

$$\begin{array}{lclcl} y(t) + \overset{*}{a}_1 y(t-1) + \ldots + \overset{*}{a}_{n_a} y(t-\overset{*}{n}_a) & = & \overset{*}{b}_1 u(t-\overset{*}{n}_r) + \ldots + \overset{*}{b}_{n_b} u(t-\overset{*}{n}_b-\overset{*}{n}_r+1) & + & n(t), \\ \textit{autoregressive}\ \text{part} & = & \textit{exogenous}\ \text{part} & + & \text{noise} \end{array}$$

where the successive values of the noise $n(t)$ (t = 1, 2, ...) are realizations of a *sequence of independent random variables* $\{\varepsilon(t)\}$. The star is used to indicate "true" values of the parameters and n_r is the input-output delay ($n_r \geq 1$ for a discretized continuous-time physical system, and we shall often assume in what follows that $n_r = 1$ to simplify notation). This corresponds to an *AutoRegressive with eXogenous variable* structure (or *ARX*). Once n_a, n_b and n_r have been chosen, the unknown parameters to be estimated are

$$\mathbf{p} = (a_1, \ldots , a_{n_a}, b_1, \ldots , b_{n_b})^T.$$

For models obtained by discretization of an nth-order continuous-time LI model, the inputs of which are maintained constant between sampling times (by a zero-order hold), n_a and n_b are equal to n, which simplifies characterization.

This very simple structure may turn out not be flexible enough to describe the properties of the perturbation. This may be remedied by using as the noise $n(t)$ a linear combination of the successive realizations of $\varepsilon(t)$, called a *Moving Average* (or *MA*), giving

$$\begin{aligned} y(t) + \overset{*}{a}_1 y(t-1) + \ldots + \overset{*}{a}_{n_a} y(t-\overset{*}{n}_a) &= \overset{*}{b}_1 u(t-\overset{*}{n}_r) + \ldots + \overset{*}{b}_{n_b} u(t-\overset{*}{n}_b-\overset{*}{n}_r+1) \\ &\quad + \varepsilon(t) + \overset{*}{c}_1 \varepsilon(t-1) + \ldots + \overset{*}{c}_{n_c} \varepsilon(t-\overset{*}{n}_c), \end{aligned}$$

i.e.

autoregressive part = exogenous part
+ moving-average part.

Once n_a, n_b, n_c and n_r have been chosen, the unknown parameters are

$$\mathbf{p} = (a_1, \ldots , a_{n_a}, b_1, \ldots , b_{n_b}, c_1, \ldots , c_{n_c})^T.$$

Such a structure is called *ARMAX* (*AutoRegressive-Moving Average with eXogenous variable*) or *CARMA* (*Controlled AutoRegressive Moving Average*). It extends the ARMA structure to the case where controlled inputs are present (Box and Jenkins, 1976). Removing the autoregressive part of an ARMAX structure, one gets a *Finite Impulse Response* (*FIR*) structure. Sometimes, replacing $y(t)$ by $\Delta y(t) = y(t) - y(t-1)$ and $u(t)$ by $\Delta u(t) = u(t) - u(t-1)$ allows one to get rid of very slowly varying perturbations, such as offsets, by working on increments. This gives the *ARIMAX* or *CARIMA* structures. There are also non-LI variants of ARMAX structures, called *NARMAX* (Leontaridis and Billings, 1985a, 1985b; Chen and Billings, 1989), which assume that

$$y(t) = f[y(t-1), \ldots , y(t-n_a), u(t-1), \ldots , u(t-n_b), \varepsilon(t-1), \ldots , \varepsilon(t-n_c), \mathbf{p}^*] + \varepsilon(t),$$

where f is a nonlinear function, for example a polynomial or rational function.

Instead of a moving average, one may use an autoregressive noise:

$$\begin{aligned} y(t) + \overset{*}{a}_1 y(t-1) + \ldots + \overset{*}{a}_{n_a} y(t-\overset{*}{n}_a) &= \overset{*}{b}_1 u(t-\overset{*}{n}_r) + \ldots + \overset{*}{b}_{n_b} u(t-\overset{*}{n}_b-\overset{*}{n}_r+1) + n(t), \\ n(t) + \overset{*}{d}_1 n(t-1) + \ldots + \overset{*}{d}_{n_d} n(t-\overset{*}{n}_d) &= \varepsilon(t). \end{aligned}$$

Once n_a, n_b, n_d and n_r are chosen, the unknown parameters are

$$\mathbf{p} = (a_1, \ldots , a_{n_a}, b_1, \ldots , b_{n_b}, d_1, \ldots , d_{n_d})^T.$$

This corresponds to an *ARARX* structure.

One may also use an autoregressive moving average (ARMA) noise:

$$y(t) + \overset{*}{a}_1 y(t-1) + \ldots + \overset{*}{a}_{n_a} y(t-\overset{*}{n}_a) = \overset{*}{b}_1 u(t-\overset{*}{n}_r) + \ldots + \overset{*}{b}_{n_b} u(t-\overset{*}{n}_b-\overset{*}{n}_r+1) + n(t),$$
$$n(t) + \overset{*}{d}_1 n(t-1) + \ldots + \overset{*}{d}_{n_d} n(t-\overset{*}{n}_d) = \varepsilon(t) + \overset{*}{c}_1 \varepsilon(t-1) + \ldots + \overset{*}{c}_{n_c} \varepsilon(t-\overset{*}{n}_c).$$

Once n_a, n_b, n_c, n_d and n_r have been chosen, the unknown parameters are

$$\mathbf{p} = (a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b}, c_1, \ldots, c_{n_c}, d_1, \ldots, d_{n_d})^{\mathrm{T}}.$$

Such a structure is called *ARARMAX.*

Let

$$\begin{aligned} A(q, \mathbf{p}) &= 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a}, \\ B(q, \mathbf{p}) &= q^{1-n_r} (b_1 q^{-1} + \ldots + b_{n_b} q^{-n_b}), \\ C(q, \mathbf{p}) &= 1 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c}, \\ D(q, \mathbf{p}) &= 1 + d_1 q^{-1} + \ldots + d_{n_d} q^{-n_d}. \end{aligned}$$

ARARMAX structures can then be written in the condensed form

$$A(q, \mathbf{p}^*) y(t) = B(q, \mathbf{p}^*) u(t) + \frac{C(q, \mathbf{p}^*)}{D(q, \mathbf{p}^*)} \varepsilon(t),$$

where $\mathbf{p}^*$ is the true value of the parameters. ARARMAX structures contain ARX, ARARX and ARMAX structures as special cases (Figure 2.3). When $A(q, \mathbf{p}) = 1$, one gets FIR structures. When $D(q) = 1 - q^{-1}$, ARIMAX structures are obtained, which shows the integration of the moving average responsible for the I in ARIMAX.

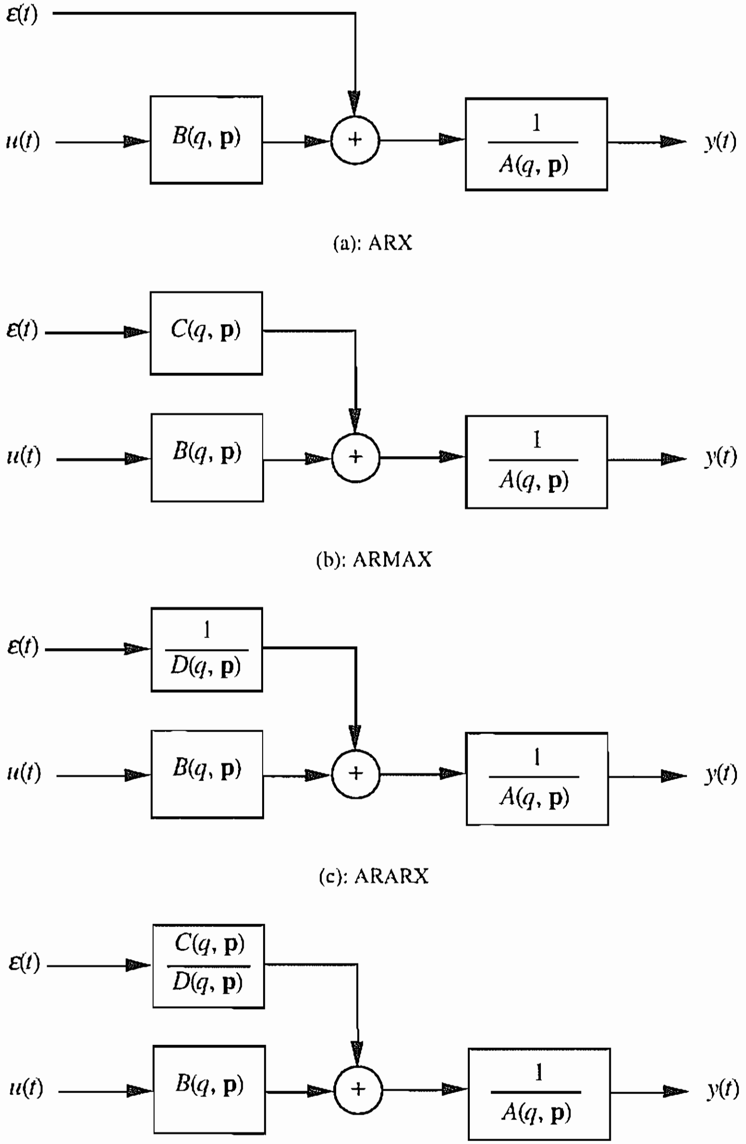
Note that these structures force the transfer functions relating the noise and input to the output to have part of their denominators in common, since

$$y(t) = \frac{B(q, \mathbf{p}^*)}{A(q, \mathbf{p}^*)} u(t) + \frac{C(q, \mathbf{p}^*)}{A(q, \mathbf{p}^*) D(q, \mathbf{p}^*)} \varepsilon(t).$$

One may prefer to parametrize these two transfer functions independently and set

$$y(t) = F(q, \mathbf{p}^*) u(t) + G(q, \mathbf{p}^*) \varepsilon(t),$$

where $F = N_F/D_F$ and $G = N_G/D_G$, with N_F, D_F, N_G and D_G polynomials in q^{-1}. One thus gets a *Box-Jenkins* structure, which may be regarded as the "most natural" parametrization of the output of a discrete-time LI system under the combined influence of deterministic and random inputs. Since the system is assumed to be LI, one can indeed easily propagate the effect of any perturbation to make it additive at the output. Moreover, the spectral factorisation theorem indicates that any rational spectrum can be obtained by passing a sequence of independent random variables $\varepsilon(t)$ identically distributed with zero mean and variance σ^2 through a filter $G(q)$ that is rational in q, stable and with a stable inverse. Insofar as the information available on the perturbations is very often of a spectral nature only, it will be possible to reproduce this type of characteristic. By suitably choosing σ, one can always require that the first element of the impulse responses of G and G^{-1} be equal to one, which will be useful for the treatments in Section 3.3.2.


$\varepsilon(t)$
$u(t)$
$B(q, \mathbf{p})$
+
$\frac{1}{A(q, \mathbf{p})}$
$y(t)$
$C(q, \mathbf{p})$
$\frac{1}{D(q, \mathbf{p})}$
$\frac{C(q, \mathbf{p})}{D(q, \mathbf{p})}$

(a): ARX

(b): ARMAX

(c): ARARX

(d): ARARMAX

Figure 2.3. ARX, ARMAX, ARARX and ARARMAX structures

REMARKS 2.3

- — The power spectrum of a stochastic process depends only on its second-order properties (mean and covariance). Signals with identical power spectra may thus differ widely in their temporal behaviour.
- — The Box-Jenkins structure may look more general than the ARARMAX structure and *a fortiori* the ARMAX structure. It is however trivial to obtain an equivalent ARMAX structure by setting $A = D_F D_G$, $B = N_F D_G$ and $C = N_G D_F$.
- — Some perturbations have a strong deterministic component and cannot be represented satisfactorily as filtered sequences of independent random variables. It may then be interesting to modify the model structure so as to incorporate this deterministic component, possibly in a parametrized form. ◊

2.5 Choice of complexity

Consider two model structures M_1 and M_2, such that with any model of structure M_1 one can associate a more complex model of structure M_2 with the same behaviour. M_2 then includes M_1, and possesses more degrees of freedom. M_1 may for example correspond to a first-order transfer function

$$H_1(s, \mathbf{p}) = \frac{p_1}{1 + p_2 s}, \ \mathbf{p} \in \mathbb{R}^2,$$

and M_2 to a second-order transfer function

$$H_2(s, \mathbf{p}) = \frac{p_1 + p_2 s}{1 + p_3 s + p_4 s^2}, \ \mathbf{p} \in \mathbb{R}^4.$$

M_1 and M_2 may also correspond to ARMAX structures of increasing complexity such that the first is a special case of the second.

The set of all input-output behaviour that can be generated by M_2 then contains all input-output behaviour that can be generated by M_1. Because of its additional degrees of freedom, M_2 can reproduce any *given* set of experimental data better than M_1. One might therefore think that the larger the number of degrees of freedom of the structure, the better the model will perform, and that the only problem to solve is a compromise between model complexity and performance. Actually, the problem is not so simple. Assume that the structure of the process is M_1 but that the experimental data are very noisy, so that the behaviour of the best model with structure M_1 fits these data very approximately. Thanks to its additional degrees of freedom, the structure M_2 may then *seem* to yield better results. Provided it is complicated enough, it may even yield a model that fits the data perfectly. However, the additional degrees of freedom in M_2 are only used here to model a particular realization of the noise corrupting the data. Should the experiment which generated the data be repeated, a different realization of this noise would result, and the best model with structure M_2 obtained from the first set of data may actually provide a much worse prediction of the behaviour of the process than the best model with structure M_1.

As will be seen in Chapter 3, criteria based on statistical considerations can be used to decide at what point increase in complexity is no longer justified. Note, however, that these criteria do not take into account the aim of the modelling, which may lead one to choose a much simpler or more complex structure than they recommend. For adaptive control, for example, one usually uses outrageously simple model structures. One may also be interested in the parameters of an LI model to describe the behaviour of a process only in a limited frequency range (*e.g.*, around some critical frequency). Filtering the input and output data with the same bandpass filter then permits elimination of irrelevant information and errors induced by offsets and high-frequency noise. (For the practical importance of such prefiltering when the model is identified for control purposes, see, *e.g.*, (Gevers, 1993).) When prior knowledge dictates a more complex phenomenological model than statistical criteria would allow, simplification of the model structure may lead to widely off-the-mark values for the phenomenological parameters, with an unrealistically optimistic evaluation of their uncertainty (Carrillo Le Roux, 1995).

Section 6.6.3 will provide tools to design experiments facilitating the choice of model structure.

2.6 Structural properties of models

Once a model structure has been chosen (or a set of structures among which a choice is to be made), its properties should be studied as independently as possible of the values taken by its parameters. As a matter of fact, this study should if possible take place before estimation of the parameters, to detect potential problems before collecting data. A property will be said to be *structural* (or *generic*) if it is true for *almost any* value of the parameters, and possibly false on a subspace of the parametric space with zero measure. Thus, a property that is true for any value of $\mathbf{p}$ not on some atypical hypersurface will be considered as structural, because the probability of randomly picking an atypical value of $\mathbf{p}$ is zero. Two structural properties are of special importance in model building, namely *identifiability* and *distinguishability*.

2.6.1 Identifiability

Consider a process and in parallel with it a model structure, the parameters of which are to be estimated according to the scheme described by Figure 1.5. Before starting data collection and parameter estimation, it seems natural to ask oneself whether one stands any chance of success, *i.e.* whether the planned measurements will contain enough information for the estimation of $\mathbf{p}$. Formulated in such vague terms, the question has no answer, so we shall consider an idealized framework (Figure 2.4) where

— the process and model have identical structure (no characterization error),
— the data are noise-free ($\mathbf{n}(t) \equiv \mathbf{0}$),
— the input $\mathbf{u}$ and measurement times can be chosen at will.

Under these conditions, it is always possible (*e.g.*, by choosing $\hat{\mathbf{p}} = \mathbf{p}^*$) to tune the parameters of the model so as to make its input-output behaviour identical to that of the process for any time and input, which will be denoted by $M(\mathbf{p}^*) = M(\hat{\mathbf{p}})$.

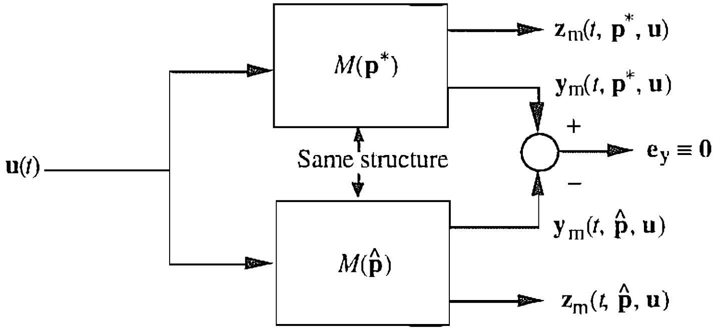

$M(\mathbf{p}^*)$
$\mathbf{z}_m(t, \mathbf{p}^*, \mathbf{u})$
$\mathbf{y}_m(t, \mathbf{p}^*, \mathbf{u})$
+
$\mathbf{u}(t)$
Same structure
$\mathbf{e}_y \equiv \mathbf{0}$
–
$\mathbf{y}_m(t, \hat{\mathbf{p}}, \mathbf{u})$
$M(\hat{\mathbf{p}})$
$\mathbf{z}_m(t, \hat{\mathbf{p}}, \mathbf{u})$

Figure 2.4. Idealized framework of structural identifiability studies

We wish to know whether this identical input-output behaviour implies that the parameters $\hat{\mathbf{p}}$ of the model equal those of the process $\mathbf{p}^*$. More precisely, parameter p_i will be *structurally globally* (or *uniquely*) *identifiable* (*s.g.i.*) if for almost any $\mathbf{p}^*$ in $\mathbb{P}$,

$$M(\hat{\mathbf{p}}) = M(\mathbf{p}^*) \Rightarrow \hat{p}_i = p_i^*.$$

We shall see below (Remarks 2.5) on a concrete example why this restriction to almost any $\mathbf{p}^*$ is necessary. The structure M will be s.g.i. if all its parameters are s.g.i.

When one cannot prove that the structure considered is globally identifiable, one may try to establish that it is at least locally. The parameter p_i will be *structurally locally identifiable* (*s.l.i.*) if for almost any $\mathbf{p}^*$ in $\mathbb{P}$, there exists a neighbourhood $\mathbb{V}(\mathbf{p}^*)$ such that

$$\hat{\mathbf{p}} \in \mathbb{V}(\mathbf{p}^*) \text{ and } M(\hat{\mathbf{p}}) = M(\mathbf{p}^*) \Rightarrow \hat{p}_i = p_i^*.$$

Local identifiability is therefore a necessary condition for global identifiability. The structure M will be s.l.i. if all its parameters are s.l.i.

The parameter p_i will be *structurally unidentifiable* (*s.u.i.*) if for almost any $\mathbf{p}^*$ in $\mathbb{P}$, there is no neighbourhood $\mathbb{V}(\mathbf{p}^*)$ such that

$$\hat{\mathbf{p}} \in \mathbb{V}(\mathbf{p}^*) \text{ and } M(\hat{\mathbf{p}}) = M(\mathbf{p}^*) \Rightarrow \hat{p}_i = p_i^*$$

The structure M will be s.u.i. if one at least of its parameters is s.u.i.

REMARKS 2.4

— Some parameters of a s.u.i. model may very well be s.l.i. or even s.g.i.

— Identifiability may depend on the numerical value taken by the parameters without the atypical region being of zero measure. It is then impossible to reach a structural conclusion (a model may be neither s.l.i. nor s.u.i.). Unless otherwise stated, for instance, $\mathbb{P} = \mathbb{R}^{n_p}$. The number of real solutions for $\hat{\mathbf{p}}$ of $M(\hat{\mathbf{p}}) = M(\mathbf{p}^*)$ may then depend on the value of $\mathbf{p}^*$, so the number of possible values for the parameters may not be a structural result. If all possible values of $\hat{\mathbf{p}}$ but one turn out to be complex, a s.l.i. structure may thus yield a unique model. See also Remark 2.9.

— The previous definitions of identifiability can readily be adapted to situations where the shape of the input $\mathbf{u}$ and the measurement times t_i ($i = 1, \ldots, n_t$) are fixed *a* *priori*. It suffices to replace $M(\hat{\mathbf{p}}) = M(\mathbf{p}^*)$ in these definitions by $\mathbf{y}^{\mathrm{m}}(\hat{\mathbf{p}}) = \mathbf{y}^{\mathrm{m}}(\mathbf{p}^*)$, where $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ stands for the vector obtained by concatenation of all available output vectors $\mathbf{y}_{\mathrm{m}}(t_i, \mathbf{p}, \mathbf{u})$, $i = 1, \dots, n_{\mathrm{t}}$. ◊

There are many methods for testing models for structural identifiability (see, *e.g.*, (Walter, 1982, 1987; Walter and Pronzato, 1995) and the references therein). Those presented in Sections 2.6.1.1 and 2.6.1.2 apply to time-invariant LI state-space structures M described by

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \mathbf{A}(\mathbf{p})\mathbf{x} + \mathbf{B}(\mathbf{p})\mathbf{u}\ ,\ \ \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}),$$
$$\mathbf{y}_{\mathrm{m}} = \mathbf{C}(\mathbf{p})\mathbf{x} + \mathbf{D}(\mathbf{p})\mathbf{u}.$$

Non-LI structures will be considered afterwards.

2.6.1.1 Laplace transform approach

This approach was initially proposed by Bellman and Åström (1970), in the context of biological modelling. After eliminating the state from the Laplace transform of the previous equations, one gets

$$\mathbf{y}_{\mathrm{m}}(s, \mathbf{p}) = \mathbf{H}_1(s, \mathbf{p})\mathbf{u}(s) + \mathbf{H}_2(s, \mathbf{p})\mathbf{x}_0(\mathbf{p}),$$

with

$$\mathbf{H}_1(s, \mathbf{p}) = \mathbf{C}(\mathbf{p})[s\mathbf{I} - \mathbf{A}(\mathbf{p})]^{-1}\mathbf{B}(\mathbf{p}) + \mathbf{D}(\mathbf{p})$$

and

$$\mathbf{H}_2(s, \mathbf{p}) = \mathbf{C}(\mathbf{p})[s\mathbf{I} - \mathbf{A}(\mathbf{p})]^{-1}.$$

$M(\hat{\mathbf{p}}) = M(\mathbf{p}^*)$ if and only if

$$\mathbf{y}_{\mathrm{m}}(s, \hat{\mathbf{p}}) - \mathbf{y}_{\mathrm{m}}(s, \mathbf{p}^*) \equiv \mathbf{0} \quad \forall\ s, \mathbf{u}(s).$$

This results in a set of equations binding $\hat{\mathbf{p}}$ and $\mathbf{p}^*$. If for almost any $\mathbf{p}^*$ this set of equations has a unique solution for $\hat{\mathbf{p}}$, M is s.g.i. If for almost any $\mathbf{p}^*$ the set of solutions is finite or denumerable, M is s.l.i. If for almost any $\mathbf{p}^*$ the set of solutions is undenumerable, M is s.u.i. Any parameter that takes the same value in all solutions is s.g.i. Any parameter that takes its values in a finite or denumerable set is s.l.i.

Writing the transfer matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ in *canonical form*, *i.e.* a form which can be written in only one way, may simplify computation considerably, because then $M(\hat{\mathbf{p}}) = M(\mathbf{p}^*)$ if and only if the *coefficients* of $\mathbf{H}_1$ and $\mathbf{H}_2$ have the same value for $\mathbf{p} = \hat{\mathbf{p}}$ and $\mathbf{p} = \mathbf{p}^*$. A canonical form is, for instance, obtained by writing down each entry of the transfer matrix as the ratio of two polynomials ordered in s, provided that the numerator and denominator are simplified by their greatest common divisor and that the coefficient of the denominator monomial with highest (or lowest) degree in s is set equal to one. Note that this simplification means that one is only dealing with the controllable and observable part of the model, a situation that will be met again in Sections 2.6.1.2 and 2.6.1.4.

EXAMPLE 2.3

Consider the (LI, non-LP) model structure defined by

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}x_1 &= -(p_1+p_2)x_1 + p_3x_2 + u, \quad x_1(0)=0, \\ \frac{\mathrm{d}}{\mathrm{d}t}x_2 &= p_1x_1 - p_3x_2, \qquad x_2(0)=0, \\ y_\mathrm{m} &= x_2. \end{aligned}$$

The Laplace transform of these equations can be written as

$$(s+p_1+p_2)x_1(s) = p_3x_2(s) + u(s),$$

$$(s+p_3)x_2(s) = p_1x_1(s),$$

$$y_\mathrm{m}(s) = x_2(s).$$

Eliminating x_1 and x_2, one gets

$$(s+p_1+p_2)(s+p_3)y_\mathrm{m}(s) = p_1p_3y_\mathrm{m}(s) + p_1u(s);$$

hence the transfer function in canonical form is

$$H_1(s, \mathbf{p}) = \frac{p_1}{s^2 + s(p_1+p_2+p_3) + p_2p_3}.$$

$M(\hat{\mathbf{p}}) = M(\mathbf{p}^*)$ is therefore equivalent to the set of equations

$$\begin{aligned} \hat{p}_1 &= \overset{*}{p}_1, \\ \hat{p}_2 + \hat{p}_3 &= \overset{*}{p}_2 + \overset{*}{p}_3, \\ \hat{p}_2\hat{p}_3 &= \overset{*}{p}_2\overset{*}{p}_3, \end{aligned}$$

which has two solutions for $\hat{\mathbf{p}}$, namely

$$\hat{\mathbf{p}}_1 = (\overset{*}{p}_1, \overset{*}{p}_2, \overset{*}{p}_3)^\mathrm{T} \quad \text{and} \quad \hat{\mathbf{p}}_2 = (\overset{*}{p}_1, \overset{*}{p}_3, \overset{*}{p}_2)^\mathrm{T}.$$

The first parameter, which takes the same value in the two solutions, is s.g.i. The other two, which each can take two values, are only s.l.i. From noise-free data, one will therefore be able to compute the true value for p_1, whereas two possible values will be obtained for p_2 and p_3. Choosing between them will be impossible without resorting to other types of measurements or other prior knowledge than assumed during the identifiability study. ◊

REMARKS 2.5

- Had we estimated the parameters of such a model with the help of one of the iterative algorithms of Section 4.3.3, we would have found either of the two possible solutions, depending on the initial value chosen. The other solution, which yields exactly the same input-output behaviour, might have been overlooked. Knowing either of them, we can now generate the other.
- Example 2.3 illustrates the necessity of including "for almost any value of $\mathbf{p}^*$" in the definitions of structural identifiability. It is clear from the expression for the transfer function that if $\mathbf{p}^*$ is in the plane of the prior feasible parameter space defined by $p_1 = 0$, the process output will be identically zero whatever the input, so p_2 and p_3 become unidentifiable. These parameters are nevertheless s.l.i., for the plane $p_1 = 0$ can be considered atypical.
- The fact that this model is not s.g.i. entails that it will not be possible to reconstruct its state $\mathbf{x}$ uniquely, solely from the knowledge of its input-output behaviour. Depending on the model selected, two possible values for x_1 will be obtained. If the vector $\mathbf{z}$ of the quantities of interest introduced in Chapter 1 depends on state variables that are not directly measured, it is therefore important to make sure that the model is s.g.i., or at least that

$$M(\hat{\mathbf{p}}) = M(\mathbf{p}^*) \Rightarrow \mathbf{z}_{\mathrm{m}}(t, \hat{\mathbf{p}}, \mathbf{u}) \equiv \mathbf{z}_{\mathrm{m}}(t, \mathbf{p}^*, \mathbf{u}).$$

- The existence of a true value of the parameters need not be assumed; $\mathbf{p}^*$ may be seen as the parameter vector of a model generating input-output behaviour that has been deemed satisfactory. The question is whether there are other models with the same structure and the same input-output behaviour.
- If the model is only to be used for the prediction or control of the process *output*, without any constraint on quantities that cannot be measured directly, then a structure that is only locally identifiable may be perfectly satisfactory (or even an unidentifiable structure, if this raises no difficulties with the estimation algorithm). ◊

2.6.1.2 Similarity transformation approach

The model generating the data is assumed to be $M(\mathbf{p}^*)$, described by

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}^* = \mathbf{A}(\mathbf{p}^*)\mathbf{x}^* + \mathbf{B}(\mathbf{p}^*)\mathbf{u}, \;\; \mathbf{x}^*(0) = \mathbf{x}_0(\mathbf{p}^*),$$
$$\mathbf{y}_{\mathrm{m}} = \mathbf{C}(\mathbf{p}^*)\mathbf{x}^* + \mathbf{D}(\mathbf{p}^*)\mathbf{u}.$$

Let $\hat{\mathbf{x}} = \mathbf{T}\mathbf{x}^*$, where $\mathbf{T}$ is the invertible matrix of a state-space similarity transformation. Then the transformed equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{x}} = \mathbf{T}\mathbf{A}(\mathbf{p}^*)\mathbf{T}^{-1}\hat{\mathbf{x}} + \mathbf{T}\mathbf{B}(\mathbf{p}^*)\mathbf{u}, \;\; \hat{\mathbf{x}}(0) = \mathbf{T}\mathbf{x}_0(\mathbf{p}^*),$$
$$\mathbf{y}_{\mathrm{m}} = \mathbf{C}(\mathbf{p}^*)\mathbf{T}^{-1}\hat{\mathbf{x}} + \mathbf{D}(\mathbf{p}^*)\mathbf{u},$$

will obviously have the same input-output behaviour as $M(\mathbf{p}^*)$. They will correspond to a model $M(\hat{\mathbf{p}})$ if and only if

$$\begin{cases} \mathbf{A}(\hat{\mathbf{p}}) = \mathbf{T}\mathbf{A}(\mathbf{p}^*)\mathbf{T}^{-1}, \\ \mathbf{B}(\hat{\mathbf{p}}) = \mathbf{T}\mathbf{B}(\mathbf{p}^*), \\ \mathbf{C}(\hat{\mathbf{p}}) = \mathbf{C}(\mathbf{p}^*)\mathbf{T}^{-1}, \\ \mathbf{D}(\hat{\mathbf{p}}) = \mathbf{D}(\mathbf{p}^*), \\ \mathbf{x}_0(\hat{\mathbf{p}}) = \mathbf{T}\mathbf{x}_0(\mathbf{p}^*), \end{cases}$$

which is a *sufficient* set of conditions for $M(\hat{\mathbf{p}}) = M(\mathbf{p}^*)$. From Kalman's algebraic equivalence theorem, this set of conditions is also *necessary*, provided that $M(\mathbf{p}^*)$ be observable and controllable. The structural identifiability of M can then be tested by looking for all solutions for $(\hat{\mathbf{p}}, \mathbf{T})$ of these equations (Berman and Schoenfeld, 1956; Glover and Willems, 1974; Walter and Lecourtier, 1981). If for almost any $\mathbf{p}^*$ the only solution is $(\hat{\mathbf{p}}, \mathbf{T}) = (\mathbf{p}^*, \mathbf{I})$, M is s.g.i. If for almost any $\mathbf{p}^*$ the set of solutions for $\hat{\mathbf{p}}$ is finite or denumerable, M is s.l.i.

EXAMPLE 2.3 (continued)

The state and observation equations correspond to

$$\mathbf{A}(\mathbf{p}) = \begin{bmatrix} -(p_1 + p_2) & p_3 \\ p_1 & -p_3 \end{bmatrix}, \quad \mathbf{B}(\mathbf{p}) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}(0) = \mathbf{0},$$

$$\mathbf{C}(\mathbf{p}) = [\ 0 \quad 1\ ], \quad \mathbf{D}(\mathbf{p}) = \mathbf{0},$$

and M is structurally controllable and observable, so the similarity transformation approach applies. Zero initial conditions bring no information on $\mathbf{T}$. Exploit first the structures of the observation and control matrices, with the notation $t_{ik} = [\mathbf{T}]_{ik}$:

$$\mathbf{C}(\hat{\mathbf{p}})\mathbf{T} = \mathbf{C}(\mathbf{p}^*) \Rightarrow t_{21} = 0,\ t_{22} = 1,$$
$$\mathbf{B}(\hat{\mathbf{p}}) = \mathbf{T}\mathbf{B}(\mathbf{p}^*) \Rightarrow t_{11} = 1,$$

so $\mathbf{T}$ can be written as

$$\mathbf{T}(\alpha) = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix}.$$

The set of all possible matrices $\mathbf{A}$ satisfies

$$\mathbf{A}(\hat{\mathbf{p}}) = \mathbf{T}(\alpha)\mathbf{A}(\mathbf{p}^*)\mathbf{T}^{-1}(\alpha)$$

$$= \begin{bmatrix} -(p_1^* + p_2^*) + \alpha p_1^* & \alpha(p_1^* + p_2^*) + p_3^* - \alpha^2 p_1^* - \alpha p_3^* \\ p_1^* & -\alpha p_1^* - p_3^* \end{bmatrix}.$$

In $\mathbf{A}(\hat{\mathbf{p}})$, the sum of the terms of the second column must be equal to zero, so

$$\alpha^2 p_1^* + \alpha(p_3^* - p_2^*) = 0.$$

This equation has two solutions for α, namely

$$\alpha = 0 \quad \Rightarrow \quad \mathbf{T} = \mathbf{I}, \;\; \hat{\mathbf{p}} = \mathbf{p}^*$$

and

$$\alpha = \frac{p_2^* - p_3^*}{p_1^*} \quad \Rightarrow \quad \mathbf{T} \neq \mathbf{I}, \;\; \hat{\mathbf{p}} = \begin{bmatrix} p_1^* \\ p_3^* \\ p_2^* \end{bmatrix}.$$

The same conclusion is reached as with the Laplace transform approach. M is s.l.i.; only p_1 is s.g.i., p_2 and p_3 can be exchanged without modifying the input-output behaviour. Even from noise-free data, it is impossible to estimate $\mathbf{p}^*$ and $\mathbf{x}^*$ uniquely, but all their possible values can now be computed. ◊

REMARK 2.6

Although the conclusion does not depend on the method used, the required computations do. Depending on the example considered, one or the other approach may turn out to be much simpler. The same holds true for methods for non-LI structures (Chappell, Godfrey and Vajda, 1990). ◊

2.6.1.3 Taylor series approach

Consider the (possibly non-LI) structure defined by

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t, \mathbf{p}], \quad \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}),$$
$$\mathbf{y}_\mathrm{m}(t, \mathbf{p}) = \mathbf{h}[\mathbf{x}(t), \mathbf{p}],$$

where $\mathbf{f}$ and $\mathbf{h}$ are assumed to be infinitely continuously differentiable. Let

$$\mathbf{a}_k(\mathbf{p}) = \lim_{t \to 0+} \frac{\mathrm{d}^k}{\mathrm{d}t^k} \mathbf{y}_\mathrm{m}(t, \mathbf{p}).$$

$M(\hat{\mathbf{p}}) = M(\mathbf{p}^*)$ implies

$$\mathbf{a}_k(\hat{\mathbf{p}}) = \mathbf{a}_k(\mathbf{p}^*), \quad k = 0, 1, \ldots$$

A *sufficient* condition for M to be s.g.i. is therefore (Pohjanpalo, 1978)

$$\mathbf{a}_k(\hat{\mathbf{p}}) = \mathbf{a}_k(\mathbf{p}^*), \; k = 0, 1, \ldots, k_\mathrm{max}, \quad \Rightarrow \quad \hat{\mathbf{p}} = \mathbf{p}^*,$$

where k_max is some positive integer, small enough for the computation to remain tractable.

EXAMPLE 2.4

Consider the matrix of all impulse responses of a LI state-space model (with $\mathbf{D} = \mathbf{0}$)

$$\mathbf{Y}_{\mathrm{m}}(t, \mathbf{p}) = \mathbf{C}(\mathbf{p})\exp[\mathbf{A}(\mathbf{p})t]\mathbf{B}(\mathbf{p}).$$

Since

$$\lim_{t \to 0+} \frac{\mathrm{d}^k}{\mathrm{d}t^k} \mathbf{Y}_{\mathrm{m}}(t, \mathbf{p}) = \mathbf{C}(\mathbf{p})\mathbf{A}^k(\mathbf{p})\mathbf{B}(\mathbf{p}),$$

the Taylor series approach amounts to testing identifiability from identity of the *Markov parameters* (Fisher, 1966; Grewal and Glover, 1976)

$$\mathbf{C}(\hat{\mathbf{p}})\mathbf{A}^k(\hat{\mathbf{p}})\mathbf{B}(\hat{\mathbf{p}}) = \mathbf{C}(\mathbf{p}^*)\mathbf{A}^k(\mathbf{p}^*)\mathbf{B}(\mathbf{p}^*),\ k = 0, 1, \dots, k_{\mathrm{max}},$$

a method that usually turns out to be more complicated than the Laplace transform and similarity transformation approaches. ◊

EXAMPLE 2.5

Consider now the unforced non-LI structure M defined by

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \begin{bmatrix} -p_1x_1 - p_2(1 - p_3x_2)x_1 \\ p_2(1 - p_3x_2)x_1 - p_4x_2 \end{bmatrix}, \quad \mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$y_{\mathrm{m}}(t, \mathbf{p}) = x_1.$$

The successive derivatives of y_{m} at $t = 0^+$ satisfy

$$a_0(\mathbf{p}) = 1,$$
$$a_1(\mathbf{p}) = -(p_1 + p_2),$$
$$a_2(\mathbf{p}) = (p_1 + p_2)^2 + p_2^2 p_3,$$
$$a_3(\mathbf{p}) = -p_2^3 p_3^2 - 4\, p_2^2 p_3(p_1 + p_2) - p_2^2 p_3 p_4 - (p_1 + p_2)^3,$$
$$\dots$$

and it is easy to show that

$$\mathbf{a}_k(\hat{\mathbf{p}}) = \mathbf{a}_k(\mathbf{p}^*),\ k = 1, 2, \dots, 5, \Rightarrow \hat{\mathbf{p}} = \mathbf{p}^*.$$

M is therefore s.g.i. ◊

REMARK 2.7

If p_3 is set to zero in Example 2.5, the model becomes LI and s.u.i. This illustrates the frequently recorded fact that LI models tend to be less identifiable than their non-LI counterparts. ◊

2.6.1.4 Local state isomorphism approach

This method applies to structures M described by

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{p}] + u(t)\mathbf{g}[\mathbf{x}(t), \mathbf{p}], \quad \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}),$$
$$\mathbf{y}_m(t, \mathbf{p}) = \mathbf{h}[\mathbf{x}(t), \mathbf{p}],$$

where **f**, **g** and **h** are analytic, u is a measurable bounded function and $M(\mathbf{p})$ is locally reduced at $\mathbf{x}_0(\mathbf{p})$ for almost any **p** (which corresponds to a notion of structural observability and structural controllability (Hermann and Krener, 1977; Sussman, 1977)). Let $\mathbf{x}^*$ be the state of $M(\mathbf{p}^*)$ and $\hat{\mathbf{x}}$ that of $M(\hat{\mathbf{p}})$. $M(\mathbf{p}^*)$ and $M(\hat{\mathbf{p}})$ will have the same input-output behaviour for any u up to some time $t_1 > 0$ if and only if there exists a local state isomorphism $\boldsymbol{\phi}$: $\mathbb{V}_0^* \rightarrow \mathbb{R}^n$, $\mathbf{x}^* \rightarrow \hat{\mathbf{x}} = \boldsymbol{\phi}(\mathbf{x}^*)$ such that *for any* $\mathbf{x}^*$ in the neighbourhood $\mathbb{V}_0^*$ of $\mathbf{x}_0(\mathbf{p}^*)$ the following conditions are met:

— $\boldsymbol{\phi}$ is a diffeomorphism:

$$\operatorname{rank} \frac{\partial \boldsymbol{\phi}(\mathbf{x})}{\partial \mathbf{x}^{\mathrm{T}}}\Big|_{\mathbf{x}=\mathbf{x}^*} = n, \tag{i}$$

— initial states correspond:

$$\boldsymbol{\phi}(\mathbf{x}_0^*) = \hat{\mathbf{x}}_0, \tag{ii}$$

— drift terms correspond:

$$\mathbf{f}(\hat{\mathbf{x}}, \hat{\mathbf{p}}) = \mathbf{f}[\boldsymbol{\phi}(\mathbf{x}^*), \hat{\mathbf{p}}] = \frac{\partial \boldsymbol{\phi}(\mathbf{x})}{\partial \mathbf{x}^{\mathrm{T}}}\Big|_{\mathbf{x}=\mathbf{x}^*} \mathbf{f}(\mathbf{x}^*, \mathbf{p}^*), \tag{iii}$$

— control terms correspond:

$$\mathbf{g}(\hat{\mathbf{x}}, \hat{\mathbf{p}}) = \mathbf{g}[\boldsymbol{\phi}(\mathbf{x}^*), \hat{\mathbf{p}}] = \frac{\partial \boldsymbol{\phi}(\mathbf{x})}{\partial \mathbf{x}^{\mathrm{T}}}\Big|_{\mathbf{x}=\mathbf{x}^*} \mathbf{g}(\mathbf{x}^*, \mathbf{p}^*), \tag{iv}$$

— observations correspond:

$$\mathbf{h}(\hat{\mathbf{x}}, \hat{\mathbf{p}}) = \mathbf{h}[\boldsymbol{\phi}(\mathbf{x}^*), \hat{\mathbf{p}}] = \mathbf{h}(\mathbf{x}^*, \mathbf{p}^*). \tag{v}$$

After checking that $M(\mathbf{p})$ is locally reduced at $\mathbf{x}_0(\mathbf{p})$, one can look for all solutions for $\hat{\mathbf{p}}$ and $\boldsymbol{\phi}$ of (i) — (v). If for almost any $\mathbf{p}^*$ the only possible solution is $\hat{\mathbf{p}} = \mathbf{p}^*$ and $\boldsymbol{\phi}(\mathbf{x}^*) = \mathbf{x}^*$, then M is s.g.i. (Vajda and Rabitz, 1989; Vadja, Godfrey and Rabitz, 1989).

The method may involve fairly complicated computation first to check that $M(\mathbf{p})$ is locally reduced and then to solve (i) — (v) for $\hat{\mathbf{p}}$ and $\boldsymbol{\phi}$. The two following results are therefore of interest.

- If there exists a value $\mathbf{p}_0$ of the parameters such that $M(\mathbf{p}_0)$ is LI and controllable and observable (which is trivial to test), then $M(\mathbf{p})$ is locally reduced at $\mathbf{x}(0) = \mathbf{0}$ for all $\mathbf{p}$ except at most for parameter vectors in a set of zero measure (Vajda, Godfrey and Rabitz, 1989).
- For polynomial models with linear observations, *i.e.* when the components of $\mathbf{f}$ and $\mathbf{g}$ are polynomials in $\mathbf{x}$ parametrized by $\mathbf{p}$ and $\mathbf{h}[\mathbf{x}(t, \mathbf{p}), \mathbf{p}] = \mathbf{C}(\mathbf{p})\mathbf{x}(t, \mathbf{p})$, ϕ can directly be written as a linear transformation $\hat{\mathbf{x}} = \mathbf{T}\mathbf{x}^*$, which drastically simplifies the calculations (Chappell, Godfrey and Vajda, 1990) and makes the transformation global, so that the positive time t_1 can be chosen arbitrarily large. Conditions (i) – (v) then become

$$\det \mathbf{T} \neq 0, \tag{i'}$$

$$\mathbf{T}\mathbf{x}_0(\mathbf{p}^*) = \mathbf{x}_0(\hat{\mathbf{p}}), \tag{ii'}$$

$$\mathbf{f}(\mathbf{T}\mathbf{x}^*, \hat{\mathbf{p}}) = \mathbf{T}\mathbf{f}(\mathbf{x}^*, \mathbf{p}^*), \tag{iii'}$$

$$\mathbf{g}(\mathbf{T}\mathbf{x}^*, \hat{\mathbf{p}}) = \mathbf{T}\mathbf{g}(\mathbf{x}^*, \mathbf{p}^*), \tag{iv'}$$

$$\mathbf{C}(\hat{\mathbf{p}})\mathbf{T} = \mathbf{C}(\mathbf{p}^*). \tag{v'}$$

Vajda *et al.* (1989) have applied this approach to a nonlinear model of methane pyrolysis.

For LI models also, the local isomorphism is a linear transformation, and the method reduces to the similarity transformation approach.

EXAMPLE 2.6

Consider the non-LI structure M defined by

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \begin{bmatrix} -p_1x_1 - p_2(1 - p_3x_2)x_1 \\ p_2(1 - p_3x_2)x_1 - p_4x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \quad \mathbf{x}(0) = \mathbf{0},$$

$$y_\mathrm{m}(t, \mathbf{p}) = x_2 = [\ 0 \quad 1\ ]\,\mathbf{x}.$$

It is trivial to check that when $p_3 = 0$ the structure becomes LI and structurally controllable and observable, so that M is structurally locally reduced and the local state isomorphism approach applies. Since M is polynomial, $\phi(\mathbf{x}) = \mathbf{T}\mathbf{x}$, and Conditions (i') – (v') express that $M(\mathbf{p}^*) = M(\hat{\mathbf{p}})$. Condition (ii') brings no information on $\mathbf{T}$. Conditions (iv') and (v') imply that it can be written as

$$\mathbf{T}(\alpha) = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix}.$$

Condition (iii') then translates into

$$\mathbf{f}(\mathbf{T}\mathbf{x}^*, \hat{\mathbf{p}}) = \begin{bmatrix} -\hat{p}_1(x_1^* + \alpha x_2^*) - \hat{p}_2(1 - \hat{p}_3 x_2^*)(x_1^* + \alpha x_2^*) \\ \hat{p}_2(1 - \hat{p}_3 x_2^*)(x_1^* + \alpha x_2^*) - \hat{p}_4 x_2^* \end{bmatrix}$$

$$= \mathbf{T}\mathbf{f}(\mathbf{x}^*, \mathbf{p}^*) = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -p_1^* x_1^* - p_2^*(1 - p_3^* x_2^*)x_1^* \\ p_2^*(1 - p_3^* x_2^*)x_1^* - p_4^* x_2^* \end{bmatrix}.$$

The second row of this equation is equivalent to

$$-\alpha\hat{p}_2\hat{p}_3 x_2^{*2} + (p_2^* p_3^* - \hat{p}_2\hat{p}_3)x_1^* x_2^* + (\hat{p}_2 - p_2^*)x_1^* + (p_4^* + \alpha\hat{p}_2 - \hat{p}_4)x_2^* = 0.$$

It must hold true for *any* $\mathbf{x}^*$ around $\mathbf{0}$ and almost any $\mathbf{p}^*$, so $\hat{p}_2 = p_2^*$, $\hat{p}_3 = p_3^*$, $\alpha = 0$ and $\hat{p}_4 = p_4^*$. Processing the first row in the same manner proves that $\hat{p}_1 = p_1^*$, so M is s.g.i. ◊

2.6.1.5 Use of elimination theory

Computer algebra software (Davenport, Siret and Tournier, 1987, 1993) such as AXIOM, MACSYMA, MAPLE or REDUCE can be used to obtain equations expressing $M(\hat{\mathbf{p}}) = M(\mathbf{p}^*)$ and solve them. These equations can often be put into the form of sets of polynomial equations in several unknowns, which can be transformed into triangular ones with the help of elimination theory (Buchberger, 1970; Raksanyi *et al.*, 1985; Lecourtier and Raksanyi, 1987). These sets of triangular equations can then be solved by considering a sequence of single-variable polynomials.

REMARK 2.8

Since $\mathbf{p}$ is assumed to belong to $\mathbb{R}^{n_\mathrm{p}}$, one should only consider real solutions to these equations, which complicates the matter. ◊

Differential algebra (Ritt, 1950; Fliess, 1989), in which differentiation is added to the classical axioms of algebra, makes it possible to use a similar approach to eliminate state variables. When differential input-output relations can be obtained that only involve known variables, their derivatives and the parameters to be estimated, these equations can be used to study identifiability (Ollivier, 1990; Diop and Fliess, 1991).

EXAMPLE 2.7

The following model corresponds to the Volmer-Heyrovski mechanism, used in electrochemistry to describe the production of gas or the dissolution of metals (Berthier *et al.*, 1995, 1996):

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = k_1(t)[1 - x(t)] - k_2(t)x(t),\ x(0) = \frac{k_1(0)}{k_1(0) + k_2(0)},$$

$$k_1(t) = p_1 \exp[p_2 u(t)],\ k_2(t) = p_3 \exp[p_4 u(t)],$$

$$y_\mathrm{m}(t) = k_1(t)[1 - x(t)] + k_2(t)x(t).$$

Differentiating the observation equation with respect to time, one obtains

$$\frac{\mathrm{d}y_{\mathrm{m}}}{\mathrm{d}t} = \frac{\mathrm{d}k_1}{\mathrm{d}t}(1-x) + \frac{\mathrm{d}k_2}{\mathrm{d}t}x + (k_2-k_1)\frac{\mathrm{d}x}{\mathrm{d}t}.$$

Replacing $\mathrm{d}x/\mathrm{d}t$ by its value as given by the state equation and multiplying the result by $(k_2 - k_1)$ so as to use the observation equation to eliminate x gives the input-output equation

$$(k_2-k_1)\frac{\mathrm{d}y_{\mathrm{m}}}{\mathrm{d}t} + [\frac{\mathrm{d}k_1}{\mathrm{d}t} - \frac{\mathrm{d}k_2}{\mathrm{d}t} + k_2^2 - k_1^2]\, y_{\mathrm{m}} = \frac{\mathrm{d}k_1}{\mathrm{d}t}k_2 - k_1\frac{\mathrm{d}k_2}{\mathrm{d}t} + 2k_1k_2(k_2-k_1),$$

with the initial condition

$$y_{\mathrm{m}}(0) = k_1(0)[1-x(0)] + k_2(0)x(0) = 2\,\frac{k_1(0)k_2(0)}{k_1(0)+k_2(0)}.$$

Exchanging k_1 and k_2 (*i.e.* (p_1, p_2) and (p_3, p_4)) leaves $y(0)$ unchanged and multiplies both sides of the input-output equation by (-1). The model is therefore not s.g.i. ◊

Finally, consider a model defined by a set of relations

$$r_k(\mathbf{y}_{\mathrm{m}}, \mathbf{u}, \mathbf{x}, \mathbf{p}) = 0, \;\; k = 1, \dots, n_{\mathrm{r}},$$

where the r_k's are polynomial functions of $\mathbf{u}$, $\mathbf{y}_{\mathrm{m}}$ and $\mathbf{x}$ and their derivatives with respect to time and polynomial functions of $\mathbf{p}$. In the idealized context of identifiability studies, any uniquely identifiable parameter p_k can be computed (Ljung and Glad, 1994) as the solution of a linear equation $a_k p_k = b_k$, where a_k and b_k are polynomial functions of the inputs, outputs and their derivatives with respect to time. Computing a_k and b_k can therefore be used to prove the global identifiability of p_k.

2.6.1.6 Numerical local approach

The computations required by algebraic approaches are sometimes too complex to be executed even by the most powerful computers. The following method can then be used to check whether M is (at least locally) identifiable.

— Choose some nominal value $\mathbf{p}_0$ for the parameters (by randomly picking it in $\mathbb{P}$). Simulate $M(\mathbf{p}_0)$ with a high precision to get many fictitious data $\mathbf{y}^{\mathrm{f}}$.

— Estimate $\mathbf{p}$ from $\mathbf{y}^{\mathrm{f}}$ by minimizing a quadratic cost in the output error (Chapter 3) with a second-order method such as those of Newton or Gauss-Newton (Chapter 4) initialized at $\hat{\mathbf{p}}^0 = \mathbf{p}_0$. If $\hat{\mathbf{p}}^k$ remains stable at $\mathbf{p}_0$, then M is s.l.i. If, on the other hand, the estimator is unstable, this may mean that M is s.u.i., or that $\mathbf{p}_0$ was close enough to an atypical hypersurface for the matrices inverted during the computation of $\hat{\mathbf{p}}^k$ to become numerically singular, or that the simulation was incorrect. Other nominal values $\mathbf{p}_0$ should then be picked before reaching a conclusion. Note that the Levenberg-Marquardt method (Section 4.3.3.5) cannot be used because it incorporates a regularization procedure.

More sophisticated local numerical approaches, which can be used to study the local dependencies between unidentifiable parameters, are described in (Walter, 1982, Chapter 3).

2.6.2 Distinguishability

One often hesitates between several model structures for the description of the same data. It is then natural to ask whether the measurements to be performed on the process will make it possible to decide which one is best. This question is that of the distinguishability of structures, which receives a partial answer in the same idealized framework as structural identifiability. One thus assumes (Figure 2.5) that the "process" is a model with structure M while its "model" has the structure $\hat{M}$, which now differs from M. The parameter vector associated with $\hat{M}$ will be denoted by $\hat{\mathbf{p}}$, and that associated with M by $\mathbf{p}$. The vectors $\hat{\mathbf{p}}$ and $\mathbf{p}$ are not necessarily of the same dimension. Since the process and its model no longer have the same structure, it may become impossible to tune the parameters $\hat{\mathbf{p}}$ of the model so as to obtain the same input-output behaviour as that of the process. It is this impossibility that may permit the elimination of structure $\hat{M}$ in favour of structure M.

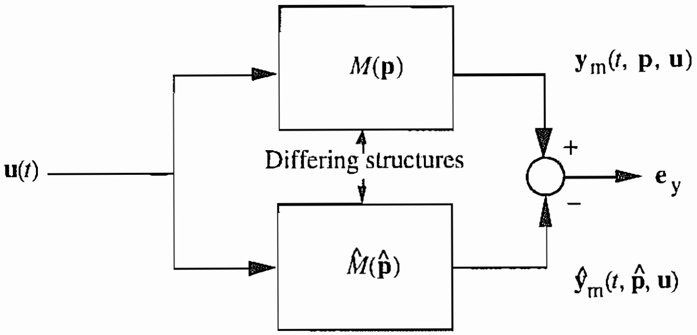

$M(\mathbf{p})$
$\mathbf{y}_m(t, \mathbf{p}, \mathbf{u})$
$\mathbf{u}(t)$
Differing structures
+
−
$\mathbf{e}_y$
$\hat{M}(\hat{\mathbf{p}})$
$\hat{\mathbf{y}}_m(t, \hat{\mathbf{p}}, \mathbf{u})$

Figure 2.5. Idealized framework of structural distinguishability studies

More precisely, $\hat{M}$ will be *structurally distinguishable (s.d.) from M* if, for almost any feasible value $\mathbf{p}$ of the parameters of M, there is no feasible value $\hat{\mathbf{p}}$ of the parameters of $\hat{M}$ such that $\hat{M}(\hat{\mathbf{p}}) = M(\mathbf{p})$.

Note the asymmetry of the previous definition. The fact that $\hat{M}$ is s.d. from M does not imply that the converse is true. One class of models may include the other (without this being obvious at first sight). Whenever $\hat{M}$ is s.d. from M and M is s.d. from $\hat{M}$, M and $\hat{M}$ are said to be s.d.

The techniques to test pairs of model structures for distinguishability are quite similar to those used for identifiability testing (Walter *et al.*, 1985). Note, however, that one now hopes to prove the non-existence of a solution for $\hat{\mathbf{p}}$, whereas in identifiability studies one hoped to prove the uniqueness of this solution.

EXAMPLE 2.8

A first model structure M is defined by

$$\frac{\mathrm{d}}{\mathrm{d}t}x_1 = -(p_1+p_2)x_1 + p_3x_2 + u, \quad x_1(0) = 0,$$
$$\frac{\mathrm{d}}{\mathrm{d}t}x_2 = p_2x_1 - p_3x_2, \quad x_2(0) = 0,$$
$$y_{\mathrm{m}} = x_1.$$

It competes with a second structure $\hat{M}$ defined by

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{x}_1 = -\hat{p}_2\hat{x}_1 + \hat{p}_3\hat{x}_2 + u, \quad \hat{x}_1(0) = 0,$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{x}_2 = \hat{p}_2\hat{x}_1 - (\hat{p}_1 + \hat{p}_3)\hat{x}_2, \quad \hat{x}_2(0) = 0,$$
$$\hat{y}_{\mathrm{m}} = \hat{x}_1.$$

The associated transfer functions, when put in the same canonical form, can respectively be written as

$$H(s, \mathbf{p}) = \frac{s + p_3}{s^2 + s(p_1 + p_2 + p_3) + p_1p_3}$$

and

$$\hat{H}(s, \hat{\mathbf{p}}) = \frac{s + \hat{p}_1 + \hat{p}_3}{s^2 + s(\hat{p}_1 + \hat{p}_2 + \hat{p}_3) + \hat{p}_1\hat{p}_2}.$$

The identity of input-output behaviour therefore translates into

$$\hat{p}_1 + \hat{p}_3 = p_3,$$
$$\hat{p}_1 + \hat{p}_2 + \hat{p}_3 = p_1 + p_2 + p_3,$$
$$\hat{p}_1\hat{p}_2 = p_1p_3.$$

For any $\mathbf{p}$, it is possible to find $\hat{\mathbf{p}}$ such that these equations are satisfied, and *vice versa*. M and $\hat{M}$ are therefore structurally indistinguishable. It will be impossible to know whether the structure chosen was right. A possible way of removing this ambiguity would be to monitor x_2. ◊

REMARK 2.9

As for identifiability, there are cases where no structural conclusion can be drawn. Consider for instance two model structures, with transfer functions

$$\hat{H}(s, \hat{\mathbf{p}}) = \frac{1}{s^2 + \hat{p}_1s + \hat{p}_2}$$

and

$$H(s, \mathbf{p}) = \frac{1}{(s + p_1)(s + p_2)},$$

with $\hat{\mathbf{p}}$ and $\mathbf{p}$ belonging to $\mathbb{R}^2$. If $\hat{H}(s, \hat{\mathbf{p}})$ has two real poles, $H(s, \mathbf{p})$ is indistinguishable from $\hat{H}(s, \hat{\mathbf{p}})$; otherwise, $H(s, \mathbf{p})$ is distinguishable from $\hat{H}(s, \hat{\mathbf{p}})$, because of the restriction of $\mathbf{p}$ to real values. None of these situations can *a priori* be considered atypical. ◊

2.6.3 Relationship between identifiability and distinguishability

It is easy to prove (Walter, Lecourtier and Happel, 1984) that the identifiability of two structures is neither necessary nor sufficient for their distinguishability. It suffices to show structures that are distinguishable when none is identifiable and identifiable structures that are indistinguishable. As already noted, the techniques to test structures for these two types of properties are nevertheless quite similar.

2.6.4 Chemical engineering example

The experimental set-up described by Figure 2.6 is used at Columbia University to study, with the help of transient isotopic tracing, the reaction that produces methane from carbon monoxide and hydrogen according to

$$CO + 3\ H_2 \Leftrightarrow CH_4 + H_2O.$$

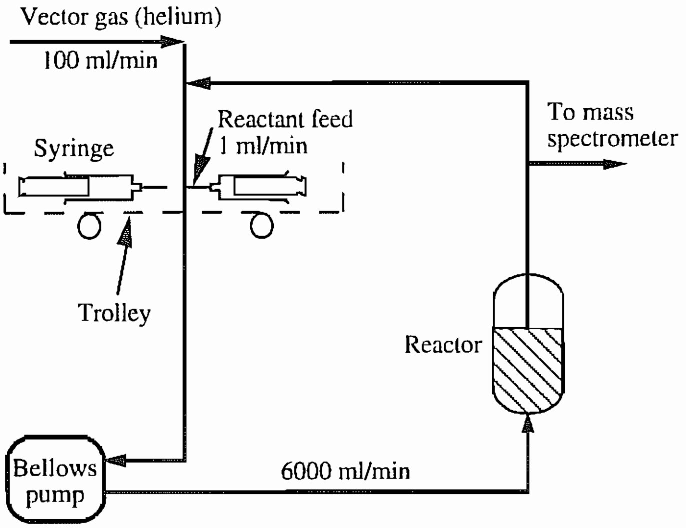


Figure 2.6. Experimental set-up in heterogeneous catalysis

Before time $t = 0$, a gaseous mixture containing carbon monoxide, methane, hydrogen and water is circulated at high speed over a nickel-based solid catalyst. The consumed carbon monoxide and hydrogen are continuously replaced with the help of a syringe, while pressure and temperature are regulated. The composition of the gaseous mixture collected at the outlet of the reactor is analysed with a mass spectrometer. When the reactor reaches a stationary state, the composition of the gaseous mixture becomes constant. This corresponds to a dynamic equilibrium, where each molecule leaving the reactor is replaced by an identical one. Then, at time $t = 0$, the syringe containing $C^{12}O$ carbon monoxide is replaced by one containing $C^{13}O$. The evolution with time of the percentage of marked carbon atoms in carbon monoxide and methane leaving the reactor is then recorded. Assuming that the reactor is stationary and that there is no isotopic effect (*i.e.* that C^{13} and C^{12} behave identically with respect to the reaction), the evolution of the tracer is linear with respect to the input, and time-invariant, even if the reaction kinetics are highly nonlinear (Happel, 1986).

Chemical prior considerations suggest two possible model structures for the description of the data. The first one M is defined by

$$\begin{aligned}
C_1\frac{\mathrm{d}}{\mathrm{d}t}x_1 &= -(V + v_e)x_1 + v_e x_2 + Vu,\\
C_2\frac{\mathrm{d}}{\mathrm{d}t}x_2 &= v_e x_1 - v_e x_2,\\
C_3\frac{\mathrm{d}}{\mathrm{d}t}x_3 &= Vx_1 - Vx_3,\\
C_4\frac{\mathrm{d}}{\mathrm{d}t}x_4 &= Vx_3 - Vx_4,\\
y_m &= x_4,
\end{aligned}$$

where the parameters to be estimated are $\mathbf{p} = (C_1, C_2, C_3, v_e)^T$, and where V and C_4 are known from independent measurements.

The second model structure $\hat{M}$ is described by

$$\begin{aligned}
C_1\frac{\mathrm{d}}{\mathrm{d}t}\hat{x}_1 &= -V\hat{x}_1 + Vu,\\
C_2\frac{\mathrm{d}}{\mathrm{d}t}\hat{x}_2 &= v_1\hat{x}_1 - v_1\hat{x}_2,\\
C_3\frac{\mathrm{d}}{\mathrm{d}t}\hat{x}_3 &= (V - v_1)\hat{x}_1 + v_1\hat{x}_2 - V\hat{x}_3,\\
C_4\frac{\mathrm{d}}{\mathrm{d}t}\hat{x}_4 &= V\hat{x}_3 - V\hat{x}_4,\\
\hat{y}_m &= \hat{x}_4,
\end{aligned}$$

where the parameters to be estimated are $\hat{\mathbf{p}} = (C_1, C_2, C_3, v_1)^T$.

The state variables x_i and $\hat{x}_i$ ($i = 1, \dots, 4$) are specific activities (percentages of labelled atoms), the C_i ($i = 1, \dots, 3$) are surface concentrations and v_e and v_1 are flow rates of carbon atoms between adsorbed species. All parameters and state variables therefore have a concrete meaning. The aim of the modelling is to determine which model structure is more adequate (discrimination or hypothesis testing) and to estimate the numerical value of the associated parameter vector. One thus hopes to detect the characteristics of the slow step of the reaction and to use this information to improve the catalyst. Before attempting to process the data, one should test whether these objectives could be met within the idealized framework of structural identifiability and distinguishability studies. With the help of the Laplace transform approach, the following conclusions can be reached (Walter *et al.*, 1986):

— M and $\hat{M}$ are structurally indistinguishable;
— M is s.l.i. but not s.g.i. (there are three different parameter vectors that correspond to exactly the same input-output behaviour);
— $\hat{M}$ is s.l.i. but not s.g.i. (there are six different parameter vectors that correspond to exactly the same input-output behaviour).

One thus knows before any measurement that it will not be possible to reach the objectives that had been initially defined. Such model structures may nevertheless be of interest, provided that their ambiguous nature is recognized and taken into account.

2.7 Conclusions

Choosing a suitable model structure is not an exact science. It involves more or less arbitrary decisions, with important consequences. Indeed, one cannot expect to do more than find the best possible model in the class thus defined. If this class is not adequate, the most sophisticated data processing will never produce a satisfactory model. One should therefore not consider characterization as an initial step to be performed once and for all, but rather as a temporary choice, bound to be questioned. The first of these questionings can take place before any actual measurement. Testing structural properties, one can detect possible defects of the model structures considered even before data are available. The *qualitative* notion of structural identifiability naturally leads to a *quantitative* question: given that the model structure under investigation is (at least locally) identifiable, what experiment should one perform to identify its parameters as precisely as possible? Similarly, when competing model structures have been proved to be distinguishable, the problem remains of effective discrimination between these structures on the basis of actual data. In both cases, the quality of the data collected is of paramount importance. The choice of optimal experimental conditions is the subject of Chapter 6.

In the next chapter, the model structure is considered as given, and we shall address the choice of the criterion to be optimized in order to find the best model in the class thus defined. When there are several possible model structures, they will be considered in turn, so as to select the simplest satisfactory one.

3 Criteria

Once the characterization has been performed, one should select the best possible model in the class thus defined, keeping in mind the final aim of the modelling. The criterion for this selection is the optimization of a scalar cost function $j(\mathbf{p})$ with respect to the model parameters $\mathbf{p}$. The optimal value of $\mathbf{p}$ (kept as the parameter estimate) will of course depend on the cost chosen, which should therefore always be specified, and justified as far as possible. Various approaches that can be employed to build criteria will be considered in this chapter. The presentation goes from the simplest and most intuitive methods to more sophisticated ones, which are more demanding in terms of prior information. When this information is unavailable (or unreliable), one may turn to the robust methods presented in Section 3.7.

Let $\mathbf{y}^{\mathrm{s}}$ be the vector of all experimental results to be used to estimate $\mathbf{p}$, and $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ the vector of the corresponding quantities computed by the model $M(\mathbf{p})$. For a parallel model, the parameters of which are to be estimated from measurements of an output vector $\mathbf{y}(t)$ ($t = t_1, \dots, t_{nt}$), these vectors can be written as

$$\mathbf{y}^{\mathrm{s}} = \begin{bmatrix} \mathbf{y}(t_1) \\ \mathbf{y}(t_2) \\ \dots \\ \mathbf{y}(t_{nt}) \end{bmatrix} \quad \text{and} \quad \mathbf{y}^{\mathrm{m}}(\mathbf{p}) = \begin{bmatrix} \mathbf{y}_{\mathrm{m}}(t_1, \mathbf{p}) \\ \mathbf{y}_{\mathrm{m}}(t_2, \mathbf{p}) \\ \dots \\ \mathbf{y}_{\mathrm{m}}(t_{nt}, \mathbf{p}) \end{bmatrix},$$

where the influence of the inputs is not made explicit to simplify notation. Due to the various perturbations acting on the system, one will usually not obtain exactly the same results when the same experiment is repeated; $\mathbf{y}^{\mathrm{s}}$ is therefore a random vector. By abuse of notation, we shall denote this random vector and its realization (vector of known numbers corresponding to actual measurements) in the same manner. We shall call any *optimizer* of j, *i.e.* any $\hat{\mathbf{p}}$ that corresponds to an optimal value of the cost function j, an *estimate* of $\mathbf{p}$ in the sense of j.

3.1 Least squares

Quadratic cost functions are by far the most commonly used, since Gauss and Legendre (Stigler, 1981), because of their intuitive appeal and relatively easy optimization (for LP models, the best estimate in the sense of a quadratic cost function can be obtained analytically, as will be seen in Chapter 4). These cost functions can be written as

$$j_{\text{ls}}(\mathbf{p}) = \mathbf{e}^{\text{T}}(\mathbf{p})\mathbf{Q}\mathbf{e}(\mathbf{p}),$$

where $\mathbf{Q}$ is a nonnegative definite symmetric weighting matrix, and $\mathbf{e}$ is a vector characterizing the error between the system and its model. The estimate of $\mathbf{p}$ in the sense of j_{ls} is given by

$$\hat{\mathbf{p}}_{\text{ls}} = \arg\min j_{\text{ls}}(\mathbf{p}).$$

Such an estimator is usually called a (*weighted*) *least-squares estimator*, or *L_2 estimator*. It can be arrived at independently of any statistical consideration, even if, as will be seen later, its use can be motivated by information (or hypotheses) on the nature of the noise acting on the system. Very often,

$$\mathbf{e}(\mathbf{p}) = \mathbf{y}^{\text{s}} - \mathbf{y}^{\text{m}}(\mathbf{p}),$$

and $\mathbf{Q}$ is chosen diagonal, so the cost can be written (after normalizing) as

$$j_{\text{ls}}(\mathbf{p}) = \frac{1}{N}\sum_{k}\sum_{t_{ik}} w_{ik}[y_k(t_{ik}) - y_{\text{m}_k}(t_{ik}, \mathbf{p})]^2,$$

where N is the total number of experimental data. Division by N permits the comparison of values of the cost obtained with different numbers of data points. The weighting coefficients w_{ik} are positive or zero and fixed *a priori*. They correspond to the diagonal entries of $\mathbf{Q}$ and may be chosen empirically. The larger w_{ik} is, and the more it will cost the model to deviate from the experimental result $y_k(t_{ik})$. The choice of the w_{ik}'s will therefore express the relative confidence in the various experimental data and the consequent importance attached to the model performance with regard to each component of $\mathbf{y}$ and associated measurement time. Thus for example

— $w_{ik} = 0$ eliminates a datum deemed insignificant,
— $w_{ik} = [y_k(t_{ik})]^{-2}$ (provided $y_k(t_{ik}) \neq 0$) makes the error relative and improves the fit of the small output values, which is of interest for example when outputs with very different amplitudes are to be fitted simultaneously.

If the data correspond to the response of the system to a step input, then

— $w_{ik} = t_{ik}$ favours the fitting of the steady state,
— $w_{ik} = 1/t_{ik}$ (provided $t_{ik} \neq 0$) favours that of the transient.

The weighting factors could be chosen iteratively. Assume, for instance, that by minimizing some initial quadratic cost, one has obtained a model with unsatisfactory behaviour in some region. By increasing the weighting factors associated with the errors in this zone, it will be possible to improve the behaviour of the model there (at the cost of its behaviour deteriorating elsewhere). By trial and error, one may then be able to correct the model response in order to reach a satisfactory compromise.

REMARK 3.1

When the data consist of *repeated measurements* at n_t times (or more generally n_t experimental conditions indexed by t), all data obtained at the same value of t can be replaced by their arithmetic mean without modifying $\hat{\mathbf{p}}_{\mathrm{ls}}$, provided that the model output $\mathbf{y}_{\mathrm{m}}(t, \mathbf{p})$ be independent of the previous measurements (which in particular does not allow consideration of models including an autoregressive part) and that the weighting matrix be suitably modified. Indeed, if $n_{\mathrm{i}}(t)$ measurements are taken under the experimental conditions indexed by t, the cost can be written as

$$j_{\mathrm{ls}}(\mathbf{p}) = \sum_{t=1}^{n_{\mathrm{t}}} \sum_{i=1}^{n_{\mathrm{i}}(t)} [\mathbf{y}_i(t) - \mathbf{y}_{\mathrm{m}}(t, \mathbf{p})]^{\mathrm{T}} \mathbf{Q}_t [\mathbf{y}_i(t) - \mathbf{y}_{\mathrm{m}}(t, \mathbf{p})],$$

and the stationarity conditions become

$$\left.\frac{\partial j}{\partial \mathbf{p}}\right|_{\hat{\mathbf{p}}_{\mathrm{ls}}} = -2 \sum_{t=1}^{n_{\mathrm{t}}} \sum_{i=1}^{n_{\mathrm{i}}(t)} \left.\frac{\partial \mathbf{y}_{\mathrm{m}}^{\mathrm{T}}(t, \mathbf{p})}{\partial \mathbf{p}}\right|_{\hat{\mathbf{p}}_{\mathrm{ls}}} \mathbf{Q}_t [\mathbf{y}_i(t) - \mathbf{y}_{\mathrm{m}}(t, \hat{\mathbf{p}}_{\mathrm{ls}})] = \mathbf{0},$$

or

$$\sum_{t=1}^{n_{\mathrm{t}}} \left.\frac{\partial \mathbf{y}_{\mathrm{m}}^{\mathrm{T}}(t, \mathbf{p})}{\partial \mathbf{p}}\right|_{\hat{\mathbf{p}}_{\mathrm{ls}}} \mathbf{Q}'_t [\mathbf{y}(t) - \mathbf{y}_{\mathrm{m}}(t, \hat{\mathbf{p}}_{\mathrm{ls}})] = \mathbf{0},$$

with

$$\mathbf{y}(t) = \frac{1}{n_{\mathrm{i}}(t)} \sum_{i=1}^{n_{\mathrm{i}}(t)} \mathbf{y}_i(t) \quad \text{and} \quad \mathbf{Q}'_t = n_{\mathrm{i}}(t) \mathbf{Q}_t. \qquad \Diamond$$

3.2 Least modulus

One should not deduce from the almost ubiquitous use of quadratic costs that they are always to be recommended. One may, for example, prefer to minimize a weighted sum of absolute values of errors, given by

$$j_{\mathrm{lm}}(\mathbf{p}) = \frac{1}{N} \sum_{k} \sum_{t_{ik}} w_{ik} \, |y_k(t_{ik}) - y_{\mathrm{m}_k}(t_{ik}, \mathbf{p})|.$$

Such cost functions, which can be traced back to the work of Galileo (in 1632!), Boscovitch and Laplace (Farebrother, 1987), penalize very large errors less than quadratic costs. The associated estimators are called (*weighted*) *least-modulus estimators*, or L_1 *estimators*. The choice of the weighting factors is inspired by the same type of consideration as for quadratic costs. For example, to work with a relative error, it suffices to set $w_{ik} = 1/|y_k(t_{ik})|$.

L_1 estimators have interesting robustness properties, as will be seen in Section 3.7. Note, however, that the estimate obtained may not be unique, even when a least-squares estimate would be. For instance, $j(p) = |p| + |p - 3|$ is at its minimum over [0, 3]. Moreover, the non-differentiable nature of j_{lm} does not allow the use of optimization techniques based on a series expansion of the cost, such as described in Section 4.3.3. For a detailed presentation of the statistical properties of L_1 estimators and of techniques to compute them, one may consult (Bloomfield and Steiger, 1983; Dodge, 1987; Gonin and Money, 1989). See also Section 4.3.5.4.

3.3 Maximum likelihood

The vector $\hat{\mathbf{p}}_{\mathrm{ml}}$ will be a maximum-likelihood estimate if it *maximizes* the cost function

$$j_{\mathrm{ml}}(\mathbf{p}) = \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}).$$

If $\mathbf{p}$ were fixed, $\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p})$ would be the probability density of the random vector $\mathbf{y}^{\mathrm{s}}$ being generated by a model with parameters $\mathbf{p}$. Here, to the contrary, $\mathbf{y}^{\mathrm{s}}$ is fixed and corresponds to the observations. Considered as a function of $\mathbf{p}$, $\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p})$ is then called the *likelihood* of $\mathbf{y}^{\mathrm{s}}$. The maximum-likelihood method looks for the value of the parameter vector $\mathbf{p}$ that gives the highest likelihood to the observed data. This approach allows one to take into account in the design of the cost the available information on the nature of the noise acting on the system.

In practice, it is often easier to look for $\hat{\mathbf{p}}_{\mathrm{ml}}$ by maximizing the *log-likelihood* function

$$j_{\mathrm{ml}}(\mathbf{p}) = \ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}),$$

which yields the same estimate since the logarithm function is monotonically increasing.

EXAMPLE 3.1: repeated observations of a Gaussian variable

Consider a system with an experimentally observed scalar output $y(t_i)$ $(i = 1, \ldots, n_t)$. Assume that it has been possible to repeat observations at each time t_i to estimate the characteristics of the measurement noise, and that the observations at time t_i are independently identically distributed (i.i.d.) according to a Gaussian law with mean μ_i and variance σ_i^2. We wish to estimate μ_i and σ_i^2 in the maximum-likelihood sense. The set of all available data for the ith time is

$$\mathbf{y}^{\mathrm{s}}(t_i) = [y_1(t_i), y_2(t_i), \ldots, y_n(t_i)]^{\mathrm{T}},$$

where $y_k(t_i)$ is the result of the kth measurement at time t_i. Since $y(t_i)$ is assumed to be distributed $\mathcal{N}(\mu_i, \sigma_i^2)$, the probability density of $y_k(t_i)$ is given by

$$\pi_{\mathrm{y}}[y_k(t_i)|\mu_i, \sigma_i^2] = \pi_{\mathrm{y}}[y_k(t_i)|\mathbf{p}] = (2\pi\sigma_i^2)^{-1/2} \exp\ \{ -\frac{1}{2}\ [\frac{y_k(t_i) - \mu_i}{\sigma_i}]^2 \}.$$

Since the n observations at time t_i are assumed to be independent, their joint probability density is the product of the probability density of each of them, *i.e.*

$$\pi_{\mathrm{y}}[\mathbf{y}^{\mathrm{s}}(t_i)|\mathbf{p}] = \prod_{k=1}^{n} \pi_{\mathrm{y}}[y_k(t_i)|\mathbf{p}] = \prod_{k=1}^{n} (2\pi\sigma_i^2)^{-1/2} \exp\ \{ -\frac{1}{2}\ [\frac{y_k(t_i) - \mu_i}{\sigma_i}]^2 \}.$$

The log-likelihood is therefore

$$j_{\mathrm{ml}}(\mathbf{p}) = \ln \pi_{\mathrm{y}}[\mathbf{y}^{\mathrm{s}}(t_i)|\mathbf{p}] = -\frac{n}{2}\ln(2\pi\sigma_i^2) - \frac{1}{2\sigma_i^2}\sum_{k=1}^{n}[y_k(t_i) - \mu_i]^2.$$

The maximum-likelihood estimate $\hat{\mathbf{p}}_{\mathrm{ml}}$ will be obtained by maximizing this cost function with respect to $\mathbf{p}$. In general, it is not possible to solve this optimization problem explicitly, and one must resort to iterative algorithms such as those described in Chapter 4. Here, however, $\hat{\mathbf{p}}_{\mathrm{ml}}$ can be obtained in explicit form. At the optimum of this unconstrained problem, the partial derivative of the cost function with respect to each of the parameters is zero (necessary first-order optimality condition), so

$$\frac{\partial}{\partial \mu_i} j_{\mathrm{ml}}|_{\hat{\mathbf{p}}_{\mathrm{ml}}} = \frac{1}{\hat{\sigma}^2_{i_{\mathrm{ml}}}}\sum_{k=1}^{n}[y_k(t_i) - \hat{\mu}_{i_{\mathrm{ml}}}] = 0,$$

$$\frac{\partial}{\partial(\sigma_i^2)} j_{\mathrm{ml}}|_{\hat{\mathbf{p}}_{\mathrm{ml}}} = -\frac{n}{2\hat{\sigma}^2_{i_{\mathrm{ml}}}} + \frac{1}{2\hat{\sigma}^4_{i_{\mathrm{ml}}}}\sum_{k=1}^{n}[y_k(t_i) - \hat{\mu}_{i_{\mathrm{ml}}}]^2 = 0.$$

The first of these equations implies

$$\hat{\mu}_{i_{\mathrm{ml}}} = \frac{1}{n}\sum_{k=1}^{n} y_k(t_i).$$

The estimator of the mean in the maximum-likelihood sense is therefore the arithmetic mean of the observations. The second equation implies

$$\hat{\sigma}^2_{i_{\mathrm{ml}}} = \frac{1}{n}\sum_{k=1}^{n}[y_k(t_i) - \hat{\mu}_{i_{\mathrm{ml}}}]^2.$$

Although presentation of the properties of maximum-likelihood estimators is deferred to Section 3.3.3, let us mention some properties of the estimators of mean and variance just derived. To simplify notation, the index i will be dropped, and the n data points will be assumed to be i.i.d. $\mathcal{N}(\mu^*, (\sigma^*)^2)$. Then (Korn and Korn, 1968) the estimator of the mean is unbiased (no systematic error):

$$\mathrm{E}\{\hat{\mu}_{\mathrm{ml}}\} = \mu^*,$$

and its variance tends to zero as n tends to infinity:

$$\mathrm{E}\{(\hat{\mu}_{\mathrm{ml}} - \mu^*)^2\} = \frac{(\sigma^*)^2}{n}.$$

On the other hand, the estimator of the variance is biased (although the bias tends to zero as n tends to infinity):

$$\mathrm{E}\{\hat{\sigma}^2_{\mathrm{ml}}\} = \frac{n-1}{n}(\sigma^*)^2.$$

It is easy to compensate for this bias by using the estimator

$$\hat{\sigma}^2_{\mathrm{u}} = \frac{n}{n-1}\,\hat{\sigma}^2_{\mathrm{ml}},$$

which also has a variance that tends to zero as n tends to infinity:

$$\mathrm{E}\{[\hat{\sigma}^2_{\mathrm{u}} - (\sigma^*)^2]^2\} = \frac{2}{n-1}(\sigma^*)^4.$$

Note that the number of repetitions must be fairly large for the estimation of μ^* and $(\sigma^*)^2$ to become accurate. With $n = 9$, for example, the standard deviations of the estimation errors are $\sigma^*/3$ for the mean μ^* and $(\sigma^*)^2/2$ for the variance $(\sigma^*)^2$. ◊

The maximum-likelihood method is the basis of a large number of estimation techniques and possesses attractive theoretical properties (Section 3.3.3). This is why its implementation will now be detailed on various examples, which illustrate the diversity of the criteria that can be produced.

3.3.1 Output-additive independent random variables

Assume that the observed outputs satisfy

$$\mathbf{y}(t_i) = \mathbf{y}_{\mathrm{m}}(t_i, \mathbf{p}^*) + \boldsymbol{\varepsilon}_i, \quad i = 1, \dots, n_{\mathrm{t}},$$

where the vector $\mathbf{y}_{\mathrm{m}}(t_i, \mathbf{p}^*)$ is the output of a deterministic model, $\mathbf{p}^*$ is the true value of the parameter vector and $\boldsymbol{\varepsilon}_i$ belongs to a sequence of independent random variables with probability density $\pi_{\boldsymbol{\varepsilon}_i}(\boldsymbol{\varepsilon}_i)$. Since the $\boldsymbol{\varepsilon}_i$'s are independent

$$\pi_{\boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \dots, \boldsymbol{\varepsilon}_{n_{\mathrm{t}}}) = \prod_{i=1}^{n_{\mathrm{t}}} \pi_{\boldsymbol{\varepsilon}_i}(\boldsymbol{\varepsilon}_i).$$

Consider the *output error*

$$\mathbf{e}_{\mathrm{y}}(t_i, \mathbf{p}) = \mathbf{y}(t_i) - \mathbf{y}_{\mathrm{m}}(t_i, \mathbf{p}).$$

For the true value of the parameters, it satisfies $\mathbf{e}_{\mathrm{y}}(t_i, \mathbf{p}^*) = \boldsymbol{\varepsilon}_i$, and, since $\mathbf{y}_{\mathrm{m}}$ is deterministic, $\pi_{\mathbf{y}_i}[\mathbf{y}(t_i)|\mathbf{p}] = \pi_{\boldsymbol{\varepsilon}_i}[\mathbf{e}_{\mathrm{y}}(t_i, \mathbf{p})]$. The likelihood of the n_{t} observations can therefore be written as

$$\pi_{\mathbf{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}) = \prod_{i=1}^{n_{\mathrm{t}}} \pi_{\boldsymbol{\varepsilon}_i}[\mathbf{e}_{\mathrm{y}}(t_i, \mathbf{p})] = \prod_{i=1}^{n_{\mathrm{t}}} \pi_{\boldsymbol{\varepsilon}_i}[\mathbf{y}(t_i) - \mathbf{y}_{\mathrm{m}}(t_i, \mathbf{p})].$$

The log-likelihood is then expressed as a sum of terms, each of which is associated with the output error for a given value of t_i,

$$\ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}) = \sum_{i=1}^{n_t} \ln \pi_{\varepsilon_i}[\mathrm{y}(t_i) - \mathrm{y}_{\mathrm{m}}(t_i, \mathbf{p})].$$

The type of probability density for the noise ε_i will dictate the type of function of the output error to be employed. Among the broad families of distributions that can be considered, Gaussian (or normal) distributions are of special importance. One justification for this phenomenon lies in the *central limit theorem*, which states that ε_i tends to be normally distributed if it results from the summation of a large number of i.i.d. errors with finite variance. A more prosaic explanation is the weight of tradition. Let us stress that other hypotheses on the noise may also be considered.

EXAMPLE 3.2: Gaussian noise with known or constant (homoscedastic) variance

Assume that the data satisfy

$$y(t_i) = y_{\mathrm{m}}(t_i, \mathbf{p}^*) + \varepsilon_i, \quad i = 1, \ldots, n_t,$$

where the ε_i's are independent $\mathcal{N}(0, \sigma_i^2)$ random variables. Assume further that the variance σ_i^2 of the noise is either constant (*i.e.* independent of i) or known for all i's. It may, for instance, have been estimated beforehand by repeated experimentation as explained in Example 3.1. One wishes to estimate $\mathbf{p}$ by maximum likelihood. According to the hypotheses on the noise,

$$\pi_{\varepsilon_i}(\varepsilon_i) = (2\pi\sigma_i^2)^{-1/2} \exp\left[-\frac{1}{2}\left(\frac{\varepsilon_i}{\sigma_i}\right)^2\right].$$

The likelihood of the n_t observations of the output is therefore given by

$$\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}) = \prod_{i=1}^{n_t} (2\pi\sigma_i^2)^{-1/2} \exp\left\{-\frac{1}{2}\left[\frac{y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})}{\sigma_i}\right]^2\right\},$$

which amounts to saying that the $y(t_i)$'s are independently distributed according to the normal law $\mathcal{N}(y_{\mathrm{m}}(t_i, \mathbf{p}), \sigma_i^2)$. The log-likelihood can be written as

$$\ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}) = (\text{term independent of } \mathbf{p}) - \frac{1}{2}\sum_{i=1}^{n_t}\left[\frac{y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})}{\sigma_i}\right]^2.$$

A maximum-likelihood estimate $\hat{\mathbf{p}}_{\mathrm{ml}}(\mathbf{y}^{\mathrm{s}})$ of $\mathbf{p}$ is therefore a maximizer of

$$j(\mathbf{p}) = -\frac{1}{2}\sum_{i=1}^{n_t}\left[\frac{y(t_i) - y_m(t_i, \mathbf{p})}{\sigma_i}\right]^2,$$

i.e. a minimizer of the quadratic cost

$$j_{ls}(\mathbf{p}) = \sum_{i=1}^{n_t} w_i\,[y(t_i) - y_m(t_i, \mathbf{p})]^2,$$

with the weights $w_i = 1/\sigma_i^2$. One thus obtains a weighted least-squares estimator. The error at a given t_i is weighted by the inverse of the variance of the associated noise; the more a measurement is corrupted by noise, the less it will influence the optimal value of the cost. Weighting thus gets a rational basis. Note that the random variables $[y(t_i) - y_m(t_i, \mathbf{p}^*)]/\sigma_i$ $(i = 1, \ldots, n_t)$ are i.i.d. $\mathcal{N}(0, 1)$. The weighting has therefore equalized the variances of the errors. This estimator can be used even if the noise does not follow a normal law, provided that its variance σ_i^2 is known. It is then called the *Gauss-Markov estimator*. For an LP model structure, it is the *Best Linear Unbiased Estimator*, or *BLUE*, in the sense of the variance of the estimates; see, *e.g.*, (Goodwin and Payne, 1977).

If all σ_i's are equal (the stationary or homoscedastic case), unweighted least-squares should be used ($w_i = 1$). The noise variance then need not be known *a priori*, and can be estimated *a posteriori* from the residuals (Example 3.4 below). ◊

EXAMPLE 3.3: Gaussian noise with unknown varying variance

Assume now that the σ_i's are unknown and may depend on i (heteroscedasticity). One may think of creating an extended parameter vector $\mathbf{p}_e = (\mathbf{p}^T, \sigma_1, \ldots, \sigma_{n_t})^T$, and estimating $\mathbf{p}_e$ by maximum likelihood. Such a strategy is however bound to fail, for $\mathbf{p}_e$ is unidentifiable since the number of extended parameters to be estimated is n_t + dim $\mathbf{p}$ when there are only n_t data points. If, on the other hand, σ_i is parametrized, *e.g.*, as (Box and Hill, 1974)

$$\sigma_i^2(a, b, \mathbf{p}) = a\,|y_m(t_i, \mathbf{p})|^b,$$

where $a > 0$ and $0 \leq b \leq 2$, then it becomes possible to estimate $\mathbf{p}_e = (\mathbf{p}^T, a, b)^T$ in the maximum-likelihood sense. Note that $b = 0$ corresponds to a constant variance (homoscedasticity), while $b = 2$ corresponds to a standard deviation proportional to the output. The likelihood of $\mathbf{y}^s$ satisfies

$$\pi_y(\mathbf{y}^s|\mathbf{p}, a, b) = \prod_{i=1}^{n_t}(2\pi a\,|y_m(t_i, \mathbf{p})|^b)^{-1/2}\exp\left\{-\frac{1}{2}\frac{[y(t_i) - y_m(t_i, \mathbf{p})]^2}{a\,|y_m(t_i, \mathbf{p})|^b}\right\},$$

and the log-likelihood of $\mathbf{y}^s$ is

$$\ln \pi_y(\mathbf{y}^s|\mathbf{p}, a, b) = -\frac{n_t}{2}\ln 2\pi - \frac{n_t}{2}\ln a - \frac{b}{2}\sum_{i=1}^{n_t} \ln |y_m(t_i, \mathbf{p})|$$

$$-\frac{1}{2a}\sum_{i=1}^{n_t} \frac{[y(t_i) - y_m(t_i, \mathbf{p})]^2}{|y_m(t_i, \mathbf{p})|^b}.$$

We must therefore *minimize* the cost

$$j(\mathbf{p}, a, b) = n_t \ln a + b\sum_{i=1}^{n_t} \ln |y_m(t_i, \mathbf{p})| + \frac{1}{a}\sum_{i=1}^{n_t} \frac{[y(t_i) - y_m(t_i, \mathbf{p})]^2}{|y_m(t_i, \mathbf{p})|^b}$$

with respect to **p**, a and b. The extended parameter a can be eliminated by noticing that

$$\left.\frac{\partial j}{\partial a}\right|_{\hat{\mathbf{p}}_{ml}, \hat{a}_{ml}, \hat{b}_{ml}} = \frac{n_t}{\hat{a}_{ml}} - \frac{1}{\hat{a}_{ml}^2}\sum_{i=1}^{n_t} \frac{[y(t_i) - y_m(t_i, \hat{\mathbf{p}}_{ml})]^2}{|y_m(t_i, \hat{\mathbf{p}}_{ml})|^{\hat{b}_{ml}}} = 0,$$

so

$$\hat{a}_{ml}(\hat{\mathbf{p}}_{ml}, \hat{b}_{ml}) = \frac{1}{n_t}\sum_{i=1}^{n_t} \frac{[y(t_i) - y_m(t_i, \hat{\mathbf{p}}_{ml})]^2}{|y_m(t_i, \hat{\mathbf{p}}_{ml})|^{\hat{b}_{ml}}}.$$

Substituting $\hat{a}_{ml}(\mathbf{p}, b)$ for a in the cost, and eliminating a term that does not depend on the parameters to be estimated, one gets

$$j_{ml}(\mathbf{p}, b) = n_t \ln \left\{ \frac{1}{n_t}\sum_{i=1}^{n_t} \frac{[y(t_i) - y_m(t_i, \mathbf{p})]^2}{|y_m(t_i, \mathbf{p})|^b} \right\} + b\sum_{i=1}^{n_t} \ln |y_m(t_i, \mathbf{p})|.$$

Although the noise is Gaussian, the cost to be minimized is *not* the intuitive quadratic cost

$$j(\mathbf{p}, b) = \frac{1}{n_t}\sum_{i=1}^{n_t} \frac{[y(t_i) - y_m(t_i, \mathbf{p})]^2}{|y_m(t_i, \mathbf{p})|^b},$$

where the errors are weighted by the inverse of the corresponding noise variance. ◊

EXAMPLE 3.4: multidimensional Gaussian noise with unknown constant covariance

Assume that several outputs of the same process are simultaneously observed, described by

$$\mathbf{y}(t_i) = \mathbf{y}_m(t_i, \mathbf{p}^*) + \boldsymbol{\varepsilon}_i, \quad i = 1, \dots, n_t,$$

where $\mathbf{y}$, $\mathbf{y}_{\text{m}}$ and $\boldsymbol{\varepsilon}_i$ are n_{y}-dimensional vectors and the $\boldsymbol{\varepsilon}_i$'s are independent Gaussian vectors with zero mean and unknown covariance $\boldsymbol{\Sigma}$ (which does not depend on i). We wish to compute maximum-likelihood estimates of $\mathbf{p}$ and $\boldsymbol{\Sigma}$. Since $\boldsymbol{\varepsilon}_i$ is distributed $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$,

$$\pi_{\varepsilon}(\boldsymbol{\varepsilon}_i) = [(2\pi)^{n_{\text{y}}} \det \boldsymbol{\Sigma}]^{-1/2} \exp\left(-\frac{1}{2}\boldsymbol{\varepsilon}_i^{\text{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\varepsilon}_i\right).$$

Taking into account the independence of the $\boldsymbol{\varepsilon}_i$'s, one can therefore write the likelihood of $\mathbf{y}^{\text{s}}$ as

$$\pi_{\text{y}}(\mathbf{y}^{\text{s}}|\mathbf{p}, \boldsymbol{\Sigma}) = \prod_{i=1}^{n_{\text{t}}} [(2\pi)^{n_{\text{y}}} \det \boldsymbol{\Sigma}]^{-1/2} \exp\left[-\frac{1}{2}\mathbf{e}_{\text{y}}(t_i, \mathbf{p})^{\text{T}}\boldsymbol{\Sigma}^{-1}\mathbf{e}_{\text{y}}(t_i, \mathbf{p})\right],$$

where

$$\mathbf{e}_{\text{y}}(t_i, \mathbf{p}) = \mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \mathbf{p}).$$

The log-likelihood of $\mathbf{y}^{\text{s}}$ satisfies

$$\ln \pi_{\text{y}}(\mathbf{y}^{\text{s}}|\mathbf{p}, \boldsymbol{\Sigma}) = -\frac{n_{\text{y}}n_{\text{t}}}{2} \ln 2\pi - \frac{n_{\text{t}}}{2} \ln \det \boldsymbol{\Sigma}$$
$$-\frac{1}{2}\sum_{i=1}^{n_{\text{t}}} [\mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \mathbf{p})]^{\text{T}}\boldsymbol{\Sigma}^{-1}[\mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \mathbf{p})].$$

If $\boldsymbol{\Sigma}$ were known, the maximum-likelihood estimate of $\mathbf{p}$ could be obtained as $\hat{\mathbf{p}}_{\text{ls}}$ which minimizes the quadratic cost

$$j_{\text{ls}}(\mathbf{p}) = \sum_{i=1}^{n_{\text{t}}} [\mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \mathbf{p})]^{\text{T}}\boldsymbol{\Sigma}^{-1}[\mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \mathbf{p})]$$

(the Gauss-Markov estimator). Since $\boldsymbol{\Sigma}$ is unknown, the log-likelihood must be maximized with respect to $\mathbf{p}$ and $\boldsymbol{\Sigma}$. Taking advantage of the first-order optimality conditions

$$\frac{\partial}{\partial \boldsymbol{\Sigma}} \ln \pi_{\text{y}}(\mathbf{y}^{\text{s}}|\mathbf{p}, \boldsymbol{\Sigma})|_{\hat{\mathbf{p}}_{\text{ml}}, \hat{\boldsymbol{\Sigma}}_{\text{ml}}} = \mathbf{0},$$

and

$$\frac{\partial}{\partial \mathbf{p}} \ln \pi_{\text{y}}(\mathbf{y}^{\text{s}}|\mathbf{p}, \boldsymbol{\Sigma})|_{\hat{\mathbf{p}}_{\text{ml}}, \hat{\boldsymbol{\Sigma}}_{\text{ml}}} = \mathbf{0},$$

and using the standard results

$$\frac{\partial}{\partial \boldsymbol{\Sigma}} \ln \det \boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{-1} \text{ and } \frac{\partial}{\partial \boldsymbol{\Sigma}} \mathbf{A}\boldsymbol{\Sigma}^{-1}\mathbf{B} = -\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{A}\boldsymbol{\Sigma}^{-1},$$

one can show (see, *e.g.*, Goodwin and Payne, 1977) that $\hat{\mathbf{p}}_{\text{ml}}$ is obtained by minimizing the cost

$$j_{\mathrm{ml}}(\mathbf{p}) = \ln \det \{\sum_{i=1}^{n_t} [\mathbf{y}(t_i) - \mathbf{y}_{\mathrm{m}}(t_i, \mathbf{p})][\mathbf{y}(t_i) - \mathbf{y}_{\mathrm{m}}(t_i, \mathbf{p})]^{\mathrm{T}}\}.$$

The estimate $\hat{\mathbf{p}}_{\mathrm{ml}}$ can then be used to compute $\hat{\mathbf{\Sigma}}_{\mathrm{ml}}$ as

$$\hat{\mathbf{\Sigma}}_{\mathrm{ml}} = \frac{1}{n_t} \sum_{i=1}^{n_t} [\mathbf{y}(t_i) - \mathbf{y}_{\mathrm{m}}(t_i, \hat{\mathbf{p}}_{\mathrm{ml}})][\mathbf{y}(t_i) - \mathbf{y}_{\mathrm{m}}(t_i, \hat{\mathbf{p}}_{\mathrm{ml}})]^{\mathrm{T}}.$$

Although $\mathbf{\Sigma}$ is unknown, it is thus possible to estimate $\mathbf{p}$ in the maximum-likelihood sense by optimizing a cost that does not depend on $\mathbf{\Sigma}$. This approach may be used to solve the problem of the relative weighting of several experimental curves without introducing more or less arbitrary weighting coefficients. It does, however, require $\mathbf{\Sigma}$ not to be singular.

When the output is scalar and ε_i is distributed $\mathcal{N}(0, \sigma^2)$, the previous result implies that $\hat{\mathbf{p}}_{\mathrm{ml}}$ is obtained by minimizing the unweighted quadratic cost

$$j_{\mathrm{ml}}(\mathbf{p}) = \sum_{i=1}^{n_t} [y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})]^2,$$

and that

$$\hat{\sigma}^2_{\mathrm{ml}} = \frac{1}{n_t} \sum_{i=1}^{n_t} [y(t_i) - y_{\mathrm{m}}(t_i, \hat{\mathbf{p}}_{\mathrm{ml}})]^2. \qquad \Diamond$$

In the vector case, the approach of Example 3.4 requires all components of $\mathbf{y}$ to be measured synchronously. Otherwise (if, *e.g.*, measurements are missing), one can easily show that if $\mathbf{\Sigma}$ is diagonal, $\hat{\mathbf{p}}_{\mathrm{ml}}$ is obtained by minimizing the cost

$$j_{\mathrm{ml}}(\mathbf{p}) = \sum_{k=1}^{n_y} \frac{n_{tk}}{2} \ln \sum_{i=1}^{n_{tk}} [y_k(t_{ik}) - y_{\mathrm{m}k}(t_{ik}, \mathbf{p})]^2 .$$

EXAMPLE 3.5: scalar Laplacian noise with known or constant standard deviation

Consider Example 3.2 again, but assume now that ε_i belongs to a sequence of independent random variables with a Laplace distribution with zero mean and known (or constant) standard deviation σ_i, so

$$\pi_{\varepsilon_i}(\varepsilon_i) = \frac{1}{\sqrt{2}\ \sigma_i} \exp\,(-\frac{\sqrt{2}\ |\varepsilon_i|}{\sigma_i}).$$

After derivations similar to those of Example 3.2, the log-likelihood of n_t observations is obtained as

$$\ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\, \mathbf{p}) = (\text{term independent of } \mathbf{p}) - \sqrt{2} \sum_{i=1}^{n_{\mathrm{t}}} \frac{|y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})|}{\sigma_i}.$$

The maximum-likelihood estimate of **p** is therefore obtained by minimizing a weighted L_1 cost (weighted least-modulus estimation)

$$j_{\mathrm{lm}}(\mathbf{p}) = \sum_{i=1}^{n_{\mathrm{t}}} w_i\, |y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})|,$$

with $w_i = 1/\sigma_i$. Such a weighting makes $w_i[y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p}^*)]$ ($i = 1, \ldots, n_{\mathrm{t}}$) i.i.d. according to a Laplace law with zero mean and unit standard deviation. When all σ_i's are identical, an unweighted least-modulus estimator should be used ($w_i = 1$). ◊

EXAMPLE 3.6: stationary uniform scalar noise

Assume that the output satisfies

$$y(t_i) = y_{\mathrm{m}}(t_i, \mathbf{p}^*) + \varepsilon_i, \;\; i = 1, \ldots, n_{\mathrm{t}},$$

where ε_i belongs to a sequence of independent random variables, uniformly distributed over the interval $[-a, a]$, *i.e.*

$$\pi_{\varepsilon}(\varepsilon_i) = \begin{cases} 1/(2a) \text{ if } |\varepsilon_i| \leq a, \\ 0 \text{ otherwise.} \end{cases}$$

The likelihood of the observations $\mathbf{y}^{\mathrm{s}}$ is then

$$\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}) = \begin{cases} 1/(2a)^{n_{\mathrm{t}}} \text{ if } |y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})| \leq a, \; i = 1, \ldots, n_{\mathrm{t}}, \\ 0 \text{ otherwise.} \end{cases}$$

Any $\hat{\mathbf{p}}$ such that $|y(t_i) - y_{\mathrm{m}}(t_i, \hat{\mathbf{p}})| \leq a$ ($i = 1, \ldots, n_{\mathrm{t}}$) is therefore a maximum-likelihood estimate of $\mathbf{p}^*$. Even is the model structure is globally identifiable, the set of all maximum-likelihood estimates of its parameters is no longer a singleton $\hat{\mathbf{p}}_{\mathrm{ml}}$. It is now a nondenumerable set of vectors, possibly empty if the hypotheses on the noise are not satisfied. Such set estimators will be considered in more detail in Section 5.4.

Should one wish to pick an element out of this set, one could use the rule

$$\hat{\mathbf{p}}_{\mathrm{mm}} = \arg \min_{\mathbf{p}} \max_{1 \leq i \leq n_{\mathrm{t}}} |y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})|.$$

This is the *minimax* or L_∞ estimator, introduced by Laplace in 1786. It makes it possible to avoid fixing the value of the error bound *a priori*. The smallest value of a consistent with the data and model structure is given by

$$\hat{a}_{\mathrm{min}} = \max_{1 \leq i \leq n_{\mathrm{t}}} |y(t_i) - y_{\mathrm{m}}(t_i, \hat{\mathbf{p}}_{\mathrm{mm}})|.$$

This approach trivially extends to the vector case. One can also consider a weighted L_∞ cost,

$$\hat{\mathbf{p}}_{\text{mm}} = \arg\min_{\mathbf{p}} \max_{1 \le i \le n_t} w_i\,|y(t_i) - y_m(t_i, \mathbf{p})|.$$

Computing $\hat{\mathbf{p}}_{\text{mm}}$ corresponds to a constrained optimization problem, since

$$\hat{\mathbf{p}}_{\text{mm}} = \arg\min_{\mathbf{p}} x,$$

under the constraints

$$x + w_i\,[y(t_i) - y_m(t_i, \mathbf{p})] \ge 0 \text{ and } x - w_i\,[y(t_i) - y_m(t_i, \mathbf{p})] \ge 0.$$

When the model structure is LP, this is a linear-programming problem, and $\hat{\mathbf{p}}_{\text{mm}}$ can be obtained recursively as the data are collected (Sections 4.3.4.1 and 5.4.1.3). ◊

3.3.2 Output-additive dependent random variables

So far, the output was assumed to be additively corrupted by a sequence of independent random variables. Now we have seen in Section 2.4 a family of discrete-time models, commonly used in practice, for which the output noise results from passing a sequence of independent random variables through a filter that destroys independence. For such models, a *prediction error* can be defined that corresponds to a sequence of independent random variables when the model parameters are equal to $\mathbf{p}^*$. This will allow the application of the results of Section 3.3.1. Consider the Box-Jenkins parametrization

$$y(t) = F(q, \mathbf{p}^*)u(t) + \eta(t),$$

where

$$\eta(t) = G(q, \mathbf{p}^*)\varepsilon(t).$$

F and G are rational functions of q^{-1}, G is *stable and with a stable inverse*, the first entry of the impulse response of G and G^{-1} is equal to one and $\varepsilon(t)$ belongs to a sequence of zero-mean independent random variables. The results for ARX and ARMAX models and their variants will be obtained by specializing F and G.

Since $\varepsilon(t)$, given by

$$\varepsilon(t) = G^{-1}(q, \mathbf{p}^*)[y(t) - F(q, \mathbf{p}^*)u(t)],$$

belongs to a sequence of independent random variables, we choose

$$e_p(t, \mathbf{p}) = y(t) - \hat{y}(t|t-1, \mathbf{p}) = G^{-1}(q, \mathbf{p})[y(t) - F(q, \mathbf{p})u(t)]$$

as the prediction error, so that

$$e_p(t, \mathbf{p}^*) = \varepsilon(t).$$

The one-step-ahead prediction of the output is given by

$$\hat{y}(t|t-1, \mathbf{p}) = G^{-1}(q, \mathbf{p})F(q, \mathbf{p})u(t) + [1 - G^{-1}(q, \mathbf{p})]y(t).$$

Since the first entry in the impulse response of G^{-1} is equal to one, this is indeed a prediction of y at time t from its past values.

REMARKS 3.2

— Computing prediction errors with the help of the previous formula would in principle require knowledge of the input and output prior to the initial time. These initial conditions are usually assumed to be zero, which is appropriate for most practical applications, provided that the filters F, G and G^{-1} are sufficiently stable. To recall the approximation made, one then speaks of *conditional* maximum likelihood (conditional on the assumptions about the initial conditions).

— Section 3.3.1 corresponds to the particular case where the prediction error is the output error, which belongs to a sequence of independent random variables if $\mathbf{p} = \mathbf{p}^*$. ◊

Since $e_\mathrm{p}(t, \mathbf{p}^*)$ belongs to a sequence of independent random variables, a (conditional) maximum-likelihood estimator of $\mathbf{p}$ will be obtained by substituting the prediction error for the output error in the approach described in Section 3.3.1. Depending on the distribution considered, various criteria will again be obtained (least squares, least modulus, minimax...).

EXAMPLE 3.7

Consider the system described by Figure 3.1, where $F(q, \mathbf{p}^*)$ is assumed to be stable with a stable inverse.

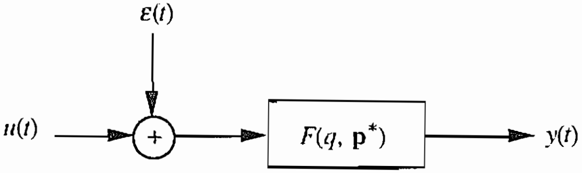


Figure 3.1. LI system, with an input-additive random perturbation

Since the system is LI, the effect of the noise can be propagated to the output to get the equivalent scheme of Figure 3.2. The one-step-ahead predictor is therefore obtained by replacing $G(q, \mathbf{p}^*)$ by $F(q, \mathbf{p}^*)$ in the previous general formula:

$$\begin{aligned}\hat{y}(t|t-1, \mathbf{p}^*) &= F^{-1}(q, \mathbf{p}^*)F(q, \mathbf{p}^*)u(t) + [1 - F^{-1}(q, \mathbf{p}^*)]y(t),\\ &= u(t) + y(t) - F^{-1}(q, \mathbf{p}^*)y(t).\end{aligned}$$

The associated prediction error does satisfy

$$e_\mathrm{p}(t, \mathbf{p}^*) = y(t) - \hat{y}(t|t-1, \mathbf{p}^*) = F^{-1}(q, \mathbf{p}^*)y(t) - u(t) = \varepsilon(t),$$

as it should. Since $F^{-1}(q, \mathbf{p}^*)y(t)$ is the noise-corrupted system input, e_p may be called the *input error* (Figure 3.3). ◊

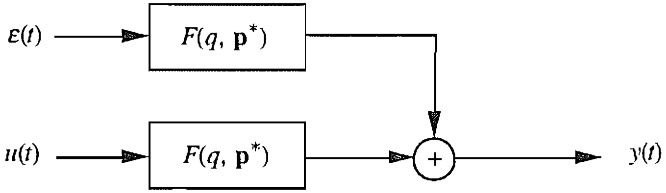


Figure 3.2. LI system equivalent to that of Figure 3.1, with an output-additive perturbation

REMARKS 3.3

— Transient errors due to ignoring nonzero initial conditions have again been neglected.

— The maximum-likelihood approach led us to take as the error a signal that would become a sequence of independent random variables if $\mathbf{p} = \mathbf{p}^*$. This could be extended to more complex models (*e.g.*, non-LI). ◊

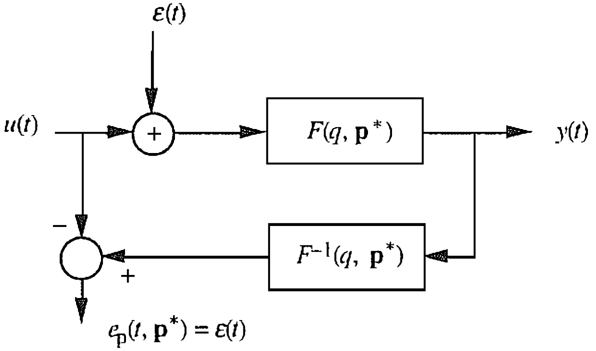

$\varepsilon(t)$
$u(t)$
+
$F(q, \mathbf{p}^*)$
$y(t)$
−
$F^{-1}(q, \mathbf{p}^*)$
+
$e_p(t, \mathbf{p}^*) = \varepsilon(t)$

Figure 3.3. Input error

3.3.3 Properties of maximum-likelihood estimators

Assume the following hypotheses are satisfied:

H1: the data are generated by a model $M(\mathbf{p}^*)$ (no characterization error);
H2: M is globally identifiable under the experimental conditions considered;
H3: the perturbations and noise influencing the data can be modelled as i.i.d. random variables, possibly passed through all or part of model $M(\mathbf{p}^*)$;
H4: the set $\{\mathbf{y}^s \mid \pi_y(\mathbf{y}^s|\mathbf{p}) > 0\}$ does not depend on $\mathbf{p}$; the second derivative of the log-likelihood $\ln \pi_y(\mathbf{y}^s|\mathbf{p})$ with respect to $\mathbf{p}$ exists, and is continuous in $\mathbf{p}$, uniformly in $\mathbf{y}^s$ (series expansion up to second order about $\mathbf{p}^*$ is possible);

H5:

$$\underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}{\mathrm{E}}\{|\frac{\partial \ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p})}{\partial p_i}|\} < \infty \text{ and } \underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}{\mathrm{E}}\{|\frac{\partial^2 \ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p})}{\partial p_i\, \partial p_k}|\} < \infty \quad \forall \mathbf{p}.$$

Maximum-likelihood estimators then have the following attractive properties (see (Fourgeaud and Fuchs, 1967, chap. 14; Kendall and Stuart, 1967; Goodwin and Payne, 1977; Sorenson, 1980) for more details).

PML1: They are *consistent*:

$$\forall\, \delta > 0,\ \mathrm{prob}(\|\hat{\mathbf{p}}_{\mathrm{ml}} - \mathbf{p}^*\| \geq \delta) \rightarrow 0 \quad \text{as} \quad n_{\mathrm{t}} \rightarrow \infty.$$

PML2: They are *asymptotically efficient*, *i.e.* there is no consistent estimator with a smaller covariance as $n_{\mathrm{t}} \rightarrow \infty$.

PML3: They are *asymptotically Gaussian and unbiased*, *i.e.* the distribution of $\hat{\mathbf{p}}_{\mathrm{ml}}$ tends to $\mathcal{N}(\mathbf{p}^*, \mathbf{F}^{-1}(\mathbf{p}^*))$ as n_{t} tends to infinity, where $\mathbf{F}$ is the *Fisher information matrix*, to be considered again in Chapter 5, given by

$$\mathbf{F}(\mathbf{p}^*) = \underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}}{\mathrm{E}} \left\{\frac{\partial}{\partial \mathbf{p}} \ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}) \frac{\partial}{\partial \mathbf{p}^{\mathrm{T}}} \ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p})\right\}\Big|_{\mathbf{p}=\mathbf{p}^*}$$
$$= - \underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}}{\mathrm{E}} \left\{\frac{\partial^2}{\partial \mathbf{p} \partial \mathbf{p}^{\mathrm{T}}} \ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p})\right\}\Big|_{\mathbf{p}=\mathbf{p}^*}.$$

REMARK 3.4

H5 can be replaced by the following more restrictive condition (Goodwin and Payne, 1977; Sorenson, 1980):

H5': $\mathbf{F}(\mathbf{p})$ is positive-definite for any $\mathbf{p}$. ◊

All previous properties are asymptotic, *i.e.* true when the number of measurements tends to infinity (by repetition of sets of measurements such that H5 or H5' is satisfied). In practice, this is never the case, and the main interest of these results is to show that maximum-likelihood estimators have satisfactory behaviour under idealized conditions. Except when the model structure is LP and the noise Gaussian (Sections 5.3.1.2 and 5.3.1.3), there are few theoretical results on the properties of maximum-likelihood estimators when data are scarce (Section 5.3.3).

A last property of maximum-likelihood estimators is of very great practical interest.

PML4: If $\mathbf{g}$ is a function of $\mathbf{p}$ only (which may correspond, *e.g.*, to a reparametrization), with $\dim \mathbf{g} \leq \dim \mathbf{p}$, then a maximum-likelihood estimate of $\mathbf{g}$ is given by $\hat{\mathbf{g}}_{\mathrm{ml}} = \mathbf{g}(\hat{\mathbf{p}}_{\mathrm{ml}})$ (*invariance principle*). ◊

To compute a maximum-likelihood estimate of any quantity that can be deduced from the knowledge of the value of $\mathbf{p}$, no other specific estimator is therefore needed. A maximum-likelihood estimate associated with a new parametrization may thus be computed from $\hat{\mathbf{p}}_{\mathrm{ml}}$. To simplify computation, it is therefore legitimate to use another parametrization than that corresponding to the actual parameters of interest.

3.3.4 Estimation of parameter distribution in a population

Consider observations $\mathbf{y}(i) = [y_1(i), \dots , y_{n_{\mathrm{i}}}(i)]^{\mathrm{T}}$ performed on various systems (or individuals) S_i ($i = 1, \dots , n_{\mathrm{s}}$) which only differ by the value of their parameters, respectively $\mathbf{p}(1), \dots , \mathbf{p}(n_{\mathrm{s}})$. The distribution $\pi_{\mathrm{p}}(\mathbf{p})$ of $\mathbf{p}$ within this population of systems is to be estimated from the observations $\mathbf{y}(1), \dots , \mathbf{y}(n_{\mathrm{s}})$. Estimating such *mixture distributions* has received much attention, because such an approach can be used in a number of situations *via* a Bayesian formulation (see, *e.g.*, (Chernov, Ososkov and Pronzato, 1992) for an application to the estimation of the disintegration points of nuclear particles from noisy observations of their trajectories). It is of special interest in biology, where knowledge of the distribution of parameters in a population of individuals makes it possible to

— estimate the parameters of each new individual in a Bayesian way, which requires few measurements since the number of data points n_{y} can then be lower than dim $\mathbf{p}$ (Thomson and Whiting, 1992),

— regulate drug concentration with the help of stochastic control methods (Jelliffe and Schumitzky, 1990; Bayard, Milman and Schumitzky, 1994; Kulcsár, Pronzato and Walter, 1994),

— optimally design data collection on the next individual in the sense of the mean over the population; see Section 6.4.2 and (Pronzato, Walter and Kulcsár, 1993).

The remainder of this section is relatively technical and could be skipped during a first reading. The likelihood of the observations $\mathbf{y}(i)$ for a distribution π_{p} of the parameters is given by

$$\pi_{\mathrm{y}}(\mathbf{y}(i)|\pi_{\mathrm{p}}) = \int_{\mathbb{P}} \pi_{\mathrm{y}}(\mathbf{y}(i)|\mathbf{p})\pi_{\mathrm{p}}(\mathbf{p})\mathrm{d}\mathbf{p}.$$

Assuming that the measurements are performed independently on the various individuals, one can also write the likelihood of the observations

$$\mathbf{y} = [\mathbf{y}(1)^{\mathrm{T}}, \dots , \mathbf{y}(n_{\mathrm{s}})^{\mathrm{T}}]^{\mathrm{T}}$$

as

$$\ell(\pi_{\mathrm{p}}) = \pi_{\mathrm{y}}(\mathbf{y}|\pi_{\mathrm{p}}) = \prod_{i=1}^{n_{\mathrm{s}}} \int_{\mathbb{P}} \pi_{\mathrm{y}}(\mathbf{y}(i)|\mathbf{p})\pi_{\mathrm{p}}(\mathbf{p})\mathrm{d}\mathbf{p}.$$

Define the *atomic likelihood vector* $\mathbf{v}(\mathbf{p})$ as the vector of the likelihoods of the observations on each individual for the value $\mathbf{p}$ of the parameters:

$$\mathbf{v}(\mathbf{p}) = [\pi_{\mathrm{y}}(\mathbf{y}(1)|\mathbf{p}), \dots , \pi_{\mathrm{y}}(\mathbf{y}(n_{\mathrm{s}})|\mathbf{p})]^{\mathrm{T}},$$

and consider the function defined by $\mathbf{p} \rightarrow \mathbf{v}(\mathbf{p})$. Its graph $\mathbb{G}$ corresponds to all possible values of the atomic likelihood vectors. By an abuse of notation, define the *mixture likelihood vector* $\mathbf{v}(\pi_{\mathrm{p}})$ as

$$\mathbf{v}(\pi_{\mathrm{p}}) = [\pi_{\mathrm{y}}(\mathbf{y}(1)|\pi_{\mathrm{p}}), \ldots , \pi_{\mathrm{y}}(\mathbf{y}(n_{\mathrm{s}})|\pi_{\mathrm{p}})]^{\mathrm{T}};$$

$\mathbf{v}(\pi_{\mathrm{p}})$ is therefore a linear combination of vectors $\mathbf{v}(\mathbf{p})$ and belongs to conv($\mathbb{G}$), the convex hull of $\mathbb{G}$. Then

$$\mathcal{V}(\pi_{\mathrm{p}}) = \prod_{i=1}^{n_{\mathrm{s}}} v_i(\pi_{\mathrm{p}}),$$

with $v_i(\pi_{\mathrm{p}})$ the ith component of $\mathbf{v}(\pi_{\mathrm{p}})$. A maximum-likelihood estimate $\hat{\pi}_{\mathrm{p}}$ of π_{p} is a maximizer of $\mathcal{V}(\pi_{\mathrm{p}})$. The convergence of $\hat{\pi}_{\mathrm{p}}$ towards the true distribution of the individual parameters $\mathbf{p}(i)$ will not be considered here (Kiefer and Wolfowitz, 1956; Leroux, 1992).

Uniqueness of $\mathbf{v}(\hat{\pi}_{\mathrm{p}})$ is achieved if $\mathbb{G}$ is compact (Lindsay, 1983). It therefore suffices that all $\pi_{\mathrm{y}}(\mathbf{y}(i)|\mathbf{p})$ belong to a closed and bounded set, which will usually be true in practice. From Caratheodory's theorem (Berger, 1979, 1987), any point on the boundary of conv($\mathbb{G}$), where $\mathbf{v}(\hat{\pi}_{\mathrm{p}})$ lies, can be written as a linear combination of at most n_{s} vectors of $\mathbb{G}$. A *discrete* optimal distribution can therefore be found, with at most n_{s} support points. The uniqueness of $\mathbf{v}(\hat{\pi}_{\mathrm{p}})$ does not imply that of $\hat{\pi}_{\mathrm{p}}$, which is obtained if $\mathbf{v}(\hat{\pi}_{\mathrm{p}})$ belongs to a support hyperplane of conv($\mathbb{G}$) that intersects $\mathbb{G}$ in a set of affinely independent vectors (Lindsay, 1983). Another condition will be given in Remark 3.5.

EXAMPLE 3.8

Consider two individuals with respective (scalar) parameter $p(1) = 1$ and $p(2) = 1.2$. Eight measurements are performed on each individual, according to

$$y_k(i) = 10 \exp(-p(i)t_k) + \varepsilon_{ik}, \ \ i = 1, 2, \ \ k = 1, \ldots , 8,$$

where the ε_{ik}'s are i.i.d. $\mathcal{N}(0, 1)$ and $t_k = 0.5k$ ($k = 1, \ldots , 8$). Figure 3.4 presents the graph $\mathbb{G}$ of the likelihood curve (solid line) and the boundary of its convex hull (dashed-dot). $\mathbf{v}(\hat{\pi}_{\mathrm{p}})$ is indicated by a cross. The locus of all $\mathbf{v}$'s such that $v_1 v_2 = v_1(\hat{\pi}_{\mathrm{p}})v_2(\hat{\pi}_{\mathrm{p}})$ is indicated by a dotted line, which shows the optimality of $\hat{\pi}_{\mathrm{p}}$. The discrete distribution $\hat{\pi}_{\mathrm{p}}$ has two support points $p^1 = 0.83$ and $p^2 = 1.19$, with respective weights 0.4928 and 0.5072. ◊

The algorithms of Chapter 4 do not apply to such infinite-dimensional problems, and we shall now indicate specific algorithms for optimizing the distribution $\hat{\pi}_{\mathrm{p}}$, which amounts to maximizing the concave function ln $\mathcal{V}(\pi_{\mathrm{p}})$ over a convex set. This problem is quite similar to approximate experiment design (Section 6.2.2).

The derivative of ln $\mathcal{V}(\pi_{\mathrm{p}})$ at π_{p}^1 in the direction π_{p}^2 can be written as

$$\phi(\pi_{\mathrm{p}}^1, \pi_{\mathrm{p}}^2) = \lim_{\alpha \to 0^+} \frac{1}{\alpha} \{\ln \mathcal{V}[(1-\alpha)\pi_{\mathrm{p}}^1 + \alpha\pi_{\mathrm{p}}^2] - \ln \mathcal{V}(\pi_{\mathrm{p}}^1)\} = \sum_{i=1}^{n_{\mathrm{s}}} \frac{\pi_{\mathrm{y}}[\mathbf{y}(i)|\pi_{\mathrm{p}}^2]}{\pi_{\mathrm{y}}[\mathbf{y}(i)|\pi_{\mathrm{p}}^1]} - n_{\mathrm{s}}.$$

If π_{p}^2 is the discrete distribution $\delta(\mathbf{p} - \mathbf{p}_{\mathrm{o}})$ that assigns a unit weight to the point $\mathbf{p}_{\mathrm{o}}$, then

$$\phi[\pi_{\mathrm{p}}^{1}, \delta(\mathbf{p} - \mathbf{p}_{\mathrm{o}})] = d(\pi_{\mathrm{p}}^{1}, \mathbf{p}_{\mathrm{o}}) - n_{\mathrm{s}},$$

with

$$d(\pi_{\mathrm{p}}^{1}, \mathbf{p}_{\mathrm{o}}) = \sum_{i=1}^{n_{\mathrm{s}}} \frac{\pi_{\mathrm{y}}[\mathbf{y}(i)|\mathbf{p}_{\mathrm{o}}]}{\pi_{\mathrm{y}}[\mathbf{y}(i)|\pi_{\mathrm{p}}^{1}]}.$$

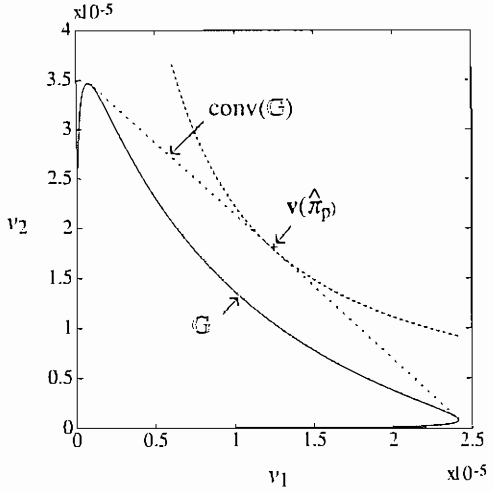


Figure 3.4. Illustration of the optimality of $\hat{\pi}_{\mathrm{p}}$ (Example 3.8)

It has been shown (Lindsay, 1983) that $\hat{\pi}_{\mathrm{p}}$ is a maximum-likelihood estimate of π_{p} if and only if

$$\max_{\mathbf{p}} d(\hat{\pi}_{\mathrm{p}}, \mathbf{p}) = n_{\mathrm{s}}$$

(compare with the Kiefer and Wolfowitz equivalence theorem, to be presented in Section 6.2.2.3).

A classical algorithm that *converges globally* (*i.e.* from any initial distribution) relies on the sequential determination of potential support points of $\hat{\pi}_{\mathrm{p}}$. It is similar to the Fedorov-Wynn algorithm presented in Section 6.2.2.4.

Step 1: Choose a positive tolerance $\varepsilon << 1$, and some discrete initial distribution

$$\pi_{\mathrm{p}}^{1}(\mathbf{p}) = \sum_{i=1}^{n} \mu_i \delta(\mathbf{p} - \mathbf{p}^i), \text{ with } \sum_{i=1}^{n} \mu_i = 1 \text{ and } \mu_i \geq 0, \ i = 1, \ldots, n.$$

Set $k = 1$.

Step 2: Compute $\mathbf{p}^{+} = \arg\max_{\mathbf{p}} d(\pi_{\mathrm{p}}^{k}, \mathbf{p})$.

Step 3: If $d(\pi_{\mathrm{p}}^{k}, \mathbf{p}^{+}) - n_{\mathrm{s}} < \varepsilon$, then stop. Else set

$$\pi_{\mathrm{p}}^{k+}(\mathbf{p}, \alpha) = (1 - \alpha)\pi_{\mathrm{p}}^{k}(\mathbf{p}) + \alpha\delta(\mathbf{p} - \mathbf{p}^{+}),$$

and find

$$\alpha^{+} = \arg \max_{\alpha \in [0,1]} \mathcal{J}(\pi_{\mathrm{p}}^{k+}).$$

Set $\pi_{\mathrm{p}}^{k+1}(\mathbf{p}) = \pi_{\mathrm{p}}^{k+}(\mathbf{p}, \alpha^{+})$, increment k by one and go to Step 2.

This procedure has the disadvantage of introducing a very large number of support points at Steps 2 and 3, whereas the optimum can be obtained with at most n_{s} points. Mallet (1986) suggests an algorithm that makes it possible to limit the number of support points of the generated distributions π_{p}^{k}. Note that the weight allocated to each new point $\mathbf{p}^{+}$ is removed from all support points of π_{p}^{k}. *Exchange algorithms* are usually more effective, which remove weight from a single support point of π_{p}^{k} and may therefore replace some support point of π_{p}^{k} by $\mathbf{p}^{+}$. A comparative study of these approaches can be found in (Böhning, 1985). See (Böhning, 1989) for monotonically convergent exchange algorithms. A detailed proof of the convergence of an exchange algorithm in the context of experiment design is given in (Huang, 1991). Finally, a particularly efficient modification is to replace the optimization with respect to α at Step 3 by a multivariable optimization of the weights μ_i allocated to the support points of π_{p}^{k+} (*i.e.* to those of π_{p}^{k} and to the point $\mathbf{p}^{+}$), under the constraints $\mu_i \geq 0$, $\sum_i \mu_i = 1$. This amounts to the maximization of a concave function over a convex set. Analytical expressions for the first and second derivative of the cost are easily obtained (Böhning, 1989), which permits the use of a constrained Newton algorithm (Section 4.3.4.5).

REMARK 3.5

A sufficient condition for the uniqueness of the distribution $\hat{\pi}_{\mathrm{p}}$ can be deduced from the necessary and sufficient condition for the optimality of $\hat{\pi}_{\mathrm{p}}$ (Mallet, 1986). Let $\tilde{\pi}_{\mathrm{p}}$ be another possibly optimal distribution. It can be shown that its support points $\tilde{\mathbf{p}}^{i}$ ($i = 1, \ldots, n$) satisfy $d(\hat{\pi}_{\mathrm{p}}, \tilde{\mathbf{p}}^{i}) = n_{\mathrm{s}}$. Let $\tilde{\mu}_i$ be the weight allocated to $\tilde{\mathbf{p}}^{i}$. Since $\mathbf{v}(\hat{\pi}_{\mathrm{p}})$ is unique, $\pi_{\mathrm{y}}[\mathbf{y}(k)|\tilde{\pi}_{\mathrm{p}}] = \pi_{\mathrm{y}}[\mathbf{y}(k)|\hat{\pi}_{\mathrm{p}}]$, ($k = 1, \ldots, n_{\mathrm{s}}$), *i.e.*

$$\sum_{i=1}^{n} \tilde{\mu}_i \, \pi_{\mathrm{y}}[\mathbf{y}(k)|\tilde{\mathbf{p}}^{i}] = \pi_{\mathrm{y}}[\mathbf{y}(k)|\hat{\pi}_{\mathrm{p}}], \quad k = 1, \ldots, n_{\mathrm{s}}.$$

It is trivial to check that the solutions of this set of equations for $\tilde{\mu}_i$ satisfy

$$\sum_{i=1}^{n} \tilde{\mu}_i = 1.$$

A necessary and sufficient uniqueness condition is therefore that the previous set of equations has a unique solution for the $\tilde{\mu}_i$'s, under the constraints $\tilde{\mu}_i \geq 0$ ($i = 1, \ldots, n$). A sufficient condition for uniqueness is thus that this system has a unique solution even when the sign constraints are not taken into account. ◊

Another family of algorithms aimed at building $\hat{\pi}_{\mathrm{p}}$ derives from the *Expectation-Maximization (EM) algorithm* (Dempster, Laird and Rubin, 1977). We shall only give an outline of the method; more details can be found in (Laird, 1978; Redner and Walker, 1984; Schumitzky, 1991).

Consider a discrete distribution π_{p} allocating weights μ_i to n support points $\mathbf{p}^{i}$ (with $n \geq n_{\mathrm{s}}$ to ensure the reachability of the optimal distribution). A transformation $\mathcal{T}$ is defined, such that $\mathcal{T}(\pi_{\rm p}) = \pi'_{\rm p}$ with support points $\mathbf{p}'^{i}$ and associated weights μ'_i given by

$$\mathbf{p}'^{i} = \arg\max_{\mathbf{p}} \sum_{k=1}^{n_{\rm s}} \pi[\mathbf{p}^{i}|\mathbf{y}(k),\ \pi_{\rm p}] \ln \pi_{\rm y}[\mathbf{y}(k)|\mathbf{p}],$$

$$\mu'_i = \frac{1}{n_{\rm s}} \sum_{k=1}^{n_{\rm s}} \pi[\mathbf{p}^{i}|\mathbf{y}(k),\ \pi_{\rm p}],$$

where

$$\pi[\mathbf{p}^{i}|\mathbf{y}(k),\ \pi_{\rm p}] = \frac{\mu_i \pi_{\rm y}[\mathbf{y}(k)|\mathbf{p}^{i}]}{\pi_{\rm y}[\mathbf{y}(k)|\pi_{\rm p}]}.$$

It can be shown (Schumitzky, 1991) that $\mathcal{V}(\pi'_{\rm p}) > \mathcal{V}(\pi_{\rm p})$, so repeatedly applying the transformation $\mathcal{T}$ monotonically increases the likelihood. A stopping criterion for the resulting optimization algorithm may be a test on the increase of the likelihood between two iterations.

REMARKS 3.6

- The major difficulty of this approach is the determination of the support points $\mathbf{p}'^{i}$ at each iteration. This difficulty disappears if these support points are imposed *a priori*, so that the transformation $\mathcal{T}$ only operates on the weights μ_i.
- The same type of algorithm applies for continuous distributions $\pi_{\rm p}$ (more precisely for distributions such that their measure is absolutely continuous with respect to the Lebesgue measure). The transformation $\mathcal{T}$ is then defined by $\mathcal{T}(\pi_{\rm p}) = \pi'_{\rm p}$, with

$$\pi'_{\rm p}(\mathbf{p}) = \frac{1}{n_{\rm s}} \sum_{k=1}^{n_{\rm s}} \pi_{\rm p}[\mathbf{p}|\mathbf{y}(k),\ \pi_{\rm p}],$$

where

$$\pi_{\rm p}[\mathbf{p}|\mathbf{y}(k),\ \pi_{\rm p}] = \frac{\pi_{\rm y}[\mathbf{y}(k)|\mathbf{p}]\pi_{\rm p}(\mathbf{p})}{\pi_{\rm y}[\mathbf{y}(k)|\pi_{\rm p}]}.$$

Again, this transformation ensures a monotonic increase of the likelihood (Schumitzky, 1991). This procedure can be initialized by some marginally informative prior law, such as a uniform distribution. Computing $\pi_{\rm y}[\mathbf{y}(k)|\pi_{\rm p}]$ requires a multidimensional numerical integration. Although the distribution obtained at each iteration is continuous, it converges towards a discrete distribution.
- No general global convergence property is known for this type of algorithm (Boyles, 1983; Wu, 1983). One is thus advised to check that the necessary and sufficient condition for the optimality of $\hat{\pi}_{\rm p}$

$$\max_{\mathbf{p}} d(\hat{\pi}_{\rm p},\ \mathbf{p}) = n_{\rm s}$$

is approximately satisfied when the algorithm stops.

- The same type of algorithm has been suggested (Silvey, Titterington and Torsney, 1978; Torsney, 1988) for experiment design (Section 6.2.2.4).
- A method for the estimation of laws relying on the Gibbs sampler has also been proposed. Its presentation exceeds the scope of this book and can be found, *e.g.*, in (Gelfand *et al.*, 1990). ◊

The approaches described in this section are *nonparametric*. They do not assume any particular structure for the distribution to be estimated. Various parametric approaches have also been proposed to estimate distributions in the context of biology (especially of pharmacokinetics), where interindividual variability is usually very high; see, *e.g.*, (Steimer *et al.*, 1984), and the NONMEM method, for *NONlinear Mixed Effect Model*, (Sheiner and Beal, 1980). These approaches estimate the mean and covariance of the distribution (which describe it completely only if it is Gaussian). Some of them can be implemented recursively, unlike the previous nonparametric approaches. The data collected on a new individual can then be used to enrich the description of the population, without having to proceed to a new estimation based on the data collected on all individuals (Mentré, 1984; Mentré, Mallet and Steimer, 1988).

3.3.5 Nonparametric description of structural errors

So far, the data were implicitly assumed to have been generated by a model $M(\mathbf{p}^*)$. The robustness of parameter estimation with respect to errors in the model structure M (which amounts to considering the presence of deterministic errors in the observations) is taken into account by the parameter-bounding approach of Section 5.4, and also considered in Section 6.6. We shall nevertheless first present a nonparametric approach that describes deviations from the assumed model as realizations of a stochastic process. The maximum-likelihood method will be used to estimate the characteristics of this process.

Assume that

$$y(\xi^i) = \mathbf{r}^{\mathrm{T}}(\xi^i)\mathbf{p}^* + \omega(\xi^i) + \varepsilon(\xi^i),$$

where ξ^i is a vector characterizing the experimental conditions under which the ith datum $y(\xi^i)$ has been collected (including, *e.g.*, the time t_i), $\mathbf{r}(\xi^i)$ is the corresponding regressor vector of an LP model (Section 4.1.1), and the $\varepsilon(\xi^i)$'s ($i = 1, \ldots, n_t$) are i.i.d. $\mathcal{N}(0, \sigma_\varepsilon^2)$. The variable ω denotes a scalar Gaussian process, assumed to be independent of the $\varepsilon(\xi^i)$'s, with zero mean and covariance defined by

$$\mathrm{cov}[\omega(\xi^i), \omega(\xi^k)] = \sigma_\omega^2 c_\omega(\xi^i, \xi^k).$$

The *deterministic* deviation from the LP model (considered as an approximation of reality) is described as a realization of the process ω. This approach is known as *kriging*, after the work of D.G. Krige (1951) on the gold deposit of the Rand, in South Africa (Matheron, 1963). Detailed presentations of the method can be found in (Sacks, Schiller and Welch, 1989; Sacks *et al.*, 1989). See also (Blight and Ott, 1975; Currin *et al.*, 1991) for a Bayesian formulation.

Various correlation structures can be used for the process ω. The deterministic nature of the structural error translates into $c_\omega(\xi, \xi) = 1$ for any ξ, so that two independent measurements under the same operating condition ξ have the same deviation $\omega(\xi)$. It seems natural to choose $c_\omega(\xi^i, \xi^k)$ as a decreasing function of $\|\xi^i - \xi^k\|_2$, such as

$$c_\omega(\xi^i, \xi^k) = \exp(-\theta \, \|\xi^i - \xi^k\|_2), \quad \theta > 0,$$

or

$$c_\omega(\xi^i, \xi^k) = \exp(-\theta \, \|\xi^i - \xi^k\|_2^2), \quad \theta > 0,$$

which corresponds to a smoother process.

A linear prediction of the mean response $\mathrm{E}_\varepsilon\{y(\xi)\}$ to operating conditions ξ which have not yet been used can be written as

$$\hat{y}(\xi) = \hat{\mathbf{c}}^{\mathrm{T}}(\xi)\mathbf{y}^{\mathrm{s}}, \quad \text{with} \quad \mathbf{y}^{\mathrm{s}} = [y(\xi^1), \ldots, y(\xi^{n_t})]^{\mathrm{T}}.$$

Denote mathematical expectation with respect to ω and ε by $\mathrm{E}\{.\}$. The mean-square error of the linear predictor is given by

$$\begin{aligned}\mathrm{MSE}(\mathbf{p}^*, \mathbf{c}, \xi) &= \mathrm{E}\{[\mathbf{c}^{\mathrm{T}}\mathbf{y}^{\mathrm{s}} - \mathrm{E}_\varepsilon\{y(\xi)\}]^2\} = \mathrm{E}\{[\mathbf{c}^{\mathrm{T}}\mathbf{y}^{\mathrm{s}} - \mathbf{r}^{\mathrm{T}}(\xi)\mathbf{p}^* - \omega(\xi)]^2\}\\ &= [\mathbf{c}^{\mathrm{T}}\mathbf{R}(\Xi)\mathbf{p}^* - \mathbf{r}^{\mathrm{T}}(\xi)\mathbf{p}^*]^2 + \mathrm{E}\{[\mathbf{c}^{\mathrm{T}}(\boldsymbol{\omega}(\Xi) + \boldsymbol{\varepsilon}(\Xi)) - \omega(\xi)]^2\},\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{\omega}(\Xi) &= [\omega(\xi^1), \ldots, \omega(\xi^{n_t})]^{\mathrm{T}},\\ \boldsymbol{\varepsilon}(\Xi) &= [\varepsilon(\xi^1), \ldots, \varepsilon(\xi^{n_t})]^{\mathrm{T}},\\ \Xi &= (\xi^1, \ldots, \xi^{n_t})^{\mathrm{T}},\end{aligned}$$

and $\mathbf{R}(\Xi)$ is the matrix with ith row equal to $\mathbf{r}^{\mathrm{T}}(\xi^i)$. In what follows, the dependence on Ξ will be omitted, and $\boldsymbol{\omega}$ and $\boldsymbol{\varepsilon}$ will stand for the vectors $\boldsymbol{\omega}(\Xi)$ and $\boldsymbol{\varepsilon}(\Xi)$, whereas $\omega(\xi)$ and $\varepsilon(\xi)$ will be scalars. One can easily show that

$$\mathrm{MSE}(\mathbf{p}^*, \mathbf{c}, \xi) = \{[\mathbf{c}^{\mathrm{T}}\mathbf{R} - \mathbf{r}^{\mathrm{T}}(\xi)]\mathbf{p}^*\}^2 + \mathbf{c}^{\mathrm{T}}\mathbf{C}_{\mathrm{y}}\mathbf{c} + \sigma_\omega^2 - 2\,\mathrm{E}_\omega\{\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}\,\omega(\xi)\},$$

with

$$\mathbf{C}_{\mathrm{y}} = \sigma_\varepsilon^2\mathbf{I}_{n_t} + \sigma_\omega^2\mathbf{C}_\omega,$$

where $\mathbf{C}_\omega$ is a matrix the (i, k) entry of which is $c_\omega(\xi^i, \xi^k)$. For the prediction to be unbiased, the following condition will be imposed

$$\mathbf{c}^{\mathrm{T}}\mathbf{R} = \mathbf{r}^{\mathrm{T}}(\xi).$$

The same condition could be obtained from a minimax-type argument: the maximum of $\mathrm{MSE}(\mathbf{p}, \mathbf{c}, \xi)$ over $\mathbf{p}$ is infinite if it is not satisfied. The minimization of $\mathrm{MSE}(\mathbf{p}, \mathbf{c}, \xi)$ with respect to $\mathbf{c}$ under the constraint $\mathbf{c}^{\mathrm{T}}\mathbf{R} = \mathbf{r}^{\mathrm{T}}(\xi)$ leads, *via* the use of the Lagrangian, to

$$\hat{y}(\xi) = \hat{\mathbf{c}}^{\mathrm{T}}(\xi)\mathbf{y}^{\mathrm{s}} = \mathbf{r}^{\mathrm{T}}(\xi)\hat{\mathbf{p}} + \mathrm{E}_\omega\{\omega(\xi)\boldsymbol{\omega}^{\mathrm{T}}\}\mathbf{C}_{\mathrm{y}}^{-1}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}\hat{\mathbf{p}}),$$

where $\hat{\mathbf{p}}$ is a least-squares estimate with the weighting matrix $\mathbf{C}_{\mathrm{y}}^{-1}$

$$\hat{\mathbf{p}} = (\mathbf{R}^{\mathrm{T}}\mathbf{C}_{\mathrm{y}}^{-1}\mathbf{R})^{-1}\mathbf{R}\mathbf{C}_{\mathrm{y}}^{-1}\mathbf{y}^{\mathrm{s}}.$$

Note that $\mathrm{E}_\omega\{\omega(\xi)\boldsymbol{\omega}^{\mathrm{T}}\}$ can be computed as an additional row of the matrix $\sigma_\omega^2\mathbf{C}_\omega$. The mean-square error of this prediction is given by

$$\hat{\mathrm{MSE}}(\xi) = \sigma_\omega^2 - \begin{bmatrix} \mathbf{r}^{\mathrm{T}}(\xi) & \mathrm{E}_\omega\{\omega(\xi)\boldsymbol{\omega}^{\mathrm{T}}\} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{R}^{\mathrm{T}} \\ \mathbf{R} & \mathbf{C}_{\mathrm{y}} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}(\xi) \\ \mathrm{E}_\omega\{\omega(\xi)\boldsymbol{\omega}\} \end{bmatrix},$$

which can be used to derive a confidence interval for the prediction.

This method can for instance be used, in the context of *computer experiments*, to predict the response of a deterministic model that can only be evaluated by executing a complex computer code (Sacks *et al.*, 1989; Currin *et al.*, 1991; Welch *et al.*, 1992)). It is then possible to predict the code output under operating conditions (input variables) ξ from runs performed under conditions ξ^i ($i = 1, \ldots, n_t$). The interest of this approach is clear when one run requires several hours of supercomputer time. If the model is deterministic, then $\sigma_\varepsilon^2 = 0$, and $\hat{\mathrm{MSE}}(\xi^i) = 0$ ($i = 1, \ldots, n_t$). The prediction is then an interpolation of the data.

EXAMPLE 3.9

Consider a deterministic system that produces the data

$$y(t) = t + 2 + \frac{30}{(1+t)^2},\quad t = 1, \ldots, 10,$$

to be described using the simpler LP model

$$y_{\mathrm{m}}(t, \mathbf{p}) = p_1 t + p_2.$$

Assume that the covariance for the systematic error between y and y_{m} can be written as

$$c_\omega(t_i, t_k) = \exp(-\theta\,|t_i - t_k|^q).$$

Figures 3.5 and 3.6 respectively illustrate the behaviour of the prediction for $\theta = 0.5$, when $q = 1$ and $q = 2$. Since there is no measurement noise, the interpolation of the data is perfect. The uncertainty on the prediction increases rapidly as soon as the data are extrapolated ($t < 1$ or $t > 10$). The higher q is, the more regular the process becomes, and the prediction is smoother in Figure 3.6 (differentiable everywhere) than in Figure 3.5 (nondifferentiable at observation points). Finally, the confidence intervals (dashed lines) are obtained here for an arbitrary choice of the parameters θ and q of the correlation function c_ω. As indicated below, these parameters can be estimated in the maximum-likelihood sense. ◊

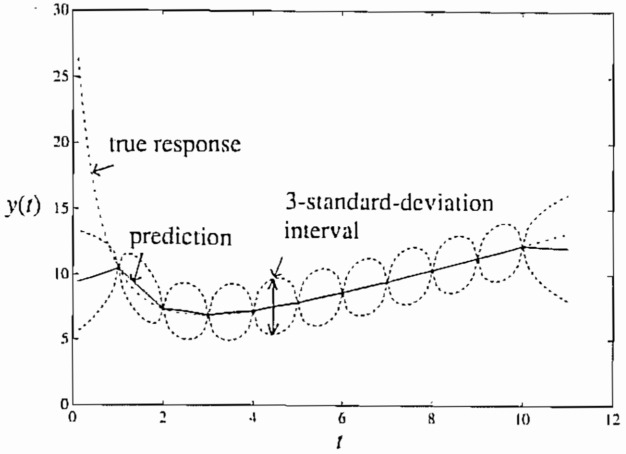


Figure 3.5. Prediction for $q = 1$: (——) prediction, (—.) true response, (- - -) three-standard-deviation interval, (*) observations

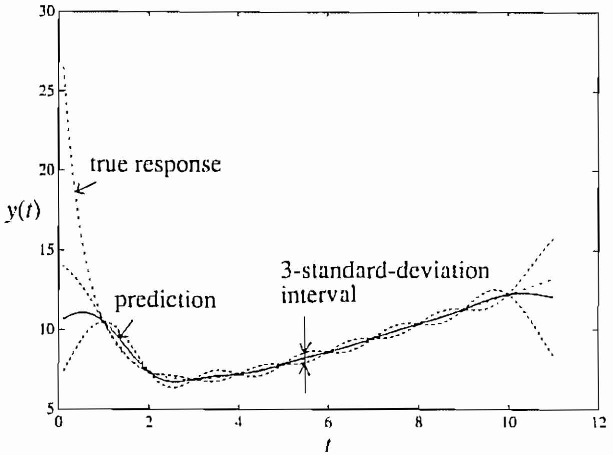


Figure 3.6. Prediction for $q = 2$: (——) prediction, (—.) true response, (- - -) three-standard-deviation interval, (*) observations

The parameters $\mathbf{p}$, σ_ε^2, σ_ω^2 as well as those of the covariance matrix $\mathbf{C}_\omega$ (θ and q in Example 3.9) can be estimated by maximum likelihood, since

$$\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}, \sigma_\varepsilon^2, \sigma_\omega^2, \mathbf{C}_\omega) = \frac{1}{[(2\pi)^{n_t} \det \mathbf{C}_{\mathrm{y}}]^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}\mathbf{p})^{\mathrm{T}}\mathbf{C}_{\mathrm{y}}^{-1}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}\mathbf{p})\right].$$

When $\sigma_\varepsilon^2 = 0$, following the same procedure as in Example 3.4, one gets

$$\hat{\mathbf{p}}(\mathbf{C}_\omega) = (\mathbf{R}^T\mathbf{C}_\omega^{-1}\mathbf{R})^{-1}\mathbf{R}^T\mathbf{C}_\omega^{-1}\mathbf{y}^s,$$

$$\hat{\sigma}_\omega^2(\mathbf{C}_\omega) = \frac{1}{n_t}[\mathbf{y}^s - \mathbf{R}\hat{\mathbf{p}}(\mathbf{C}_\omega)]^T\mathbf{C}_\omega^{-1}[\mathbf{y}^s - \mathbf{R}\hat{\mathbf{p}}(\mathbf{C}_\omega)],$$

and $\hat{\mathbf{C}}_\omega$ is obtained by minimizing ln det $[\hat{\sigma}_\omega^2(\mathbf{C}_\omega)\mathbf{C}_\omega]$. Specific algorithms for this problem can be found in (Welch *et al.*, 1992).

EXAMPLE 3.9 (continued)

A numerical optimization of ln det $(\hat{\sigma}_\omega^2(\mathbf{C}_\omega)\mathbf{C}_\omega)$ with respect to θ and q, using the Powell algorithm presented in Section 4.3.2.6, yielded $\hat{\theta} \approx 0.108$ and $\hat{q} \approx 1.99$. The constraints $\theta > 0$ and $\mathbf{C}_\omega$ positive definite were introduced through exact penalty functions (Section 3.6.2). Figure 3.7 illustrates the behaviour of the corresponding prediction. ◊

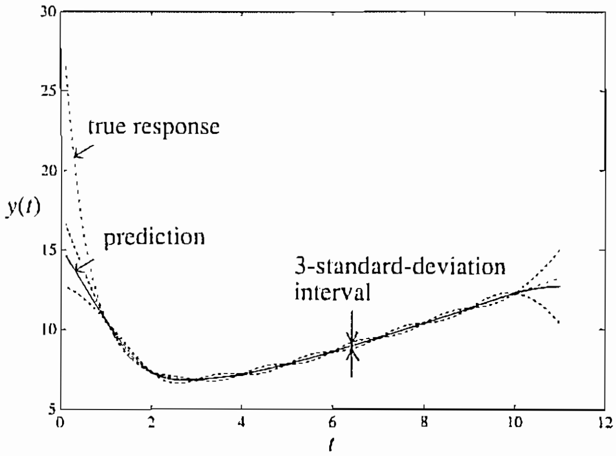


Figure 3.7. Prediction with θ and q estimated by maximum likelihood: (——) prediction, (—.) true response, (- - -) three-standard-deviation interval, (∗) observations

REMARK 3.7

Provided that the covariance matrix $\mathbf{C}_\omega$ is fixed *a priori* , the measurement (or simulation) points that define Ξ can be chosen so as to obtain the best possible prediction, for instance in the sense of one of the following costs

— the mean-square error integrated over the feasible domain ξ for ξ (Sacks and Schiller, 1988; Sacks *et al.*, 1989):

$$<\widehat{\mathrm{MSE}}> = \int_{\xi} \widehat{\mathrm{MSE}}(\xi)\, d\xi,$$

— the maximum mean-square error over ξ (Sacks and Schiller, 1988; Sacks *et al.*, 1989):

$$\widehat{\mathrm{MSE}}_{\max} = \max_{\xi \in \xi} \widehat{\mathrm{MSE}}(\xi),$$

— entropy (Shewry and Wynn, 1987, 1988).

This is a problem of *experiment design*, similar to those considered in Chapter 6. Note that the design criterion depends on the parameters of the matrix $\mathbf{C}_{\omega}$, actually unknown before the measurements have taken place. See Section 6.4 for possible approaches to overcome this contradiction partly. ◊

3.4 Complexity

Let $\mathbb{M}$ be a set of model structures that compete for the description of a given phenomenon,

$$\mathbb{M} = \{M_i,\ i = 1, \dots, n_{\mathrm{m}}\}.$$

It may correspond, for example, to structures of the same type with increasing complexity. With each of these structures is associated a parameter vector $\mathbf{p}^i$ belonging to some prior feasible set $\mathbb{P}^i$. Akaike's AIC is the most well known of the criteria that can be employed to select the model structure $\hat{M}$ and estimate its parameters on the basis of statistical considerations; see, *e.g.*, (Ljung, 1987). It suggests choosing

$$(\hat{M}, \hat{\mathbf{p}}) = \arg \min_{M_i \in \mathbb{M}} \min_{\mathbf{p}^i \in \mathbb{P}^i} j_{\mathrm{aic}}(M_i, \mathbf{p}^i),$$

where

$$j_{\mathrm{aic}}(M_i, \mathbf{p}^i) = \frac{1}{n_{\mathrm{t}}} \{-\ln [\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}^i)] + \dim \mathbf{p}^i\}.$$

When the model structure is fixed, it corresponds to maximum-likelihood estimation of its parameters. Conversely, if one hesitates between several structures, the most complex ones (in the sense of the number of parameters to be estimated and thereby the number of degrees of freedom of the model) are penalized by the term dim $\mathbf{p}$. Many other criteria for choosing the model complexity rely on this type of penalty function and only differ by the way in which dim $\mathbf{p}$ is introduced. One may quote the FPE (*Final Prediction Error*) criterion, which amounts to minimizing

$$j_{\mathrm{fpe}}(M_i, \mathbf{p}^i) = \ln \{-\ln [\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}^i)]\} + \ln \frac{1 + (\dim \mathbf{p}^i)/n_{\mathrm{t}}}{1 - (\dim \mathbf{p}^i)/n_{\mathrm{t}}},$$

and the BIC (*Bayesian Information Criterion*), which corresponds to minimizing

$$j_{\text{bic}}(M_i, \mathbf{p}^i) = \ln \{-\ln [\pi_{\text{y}}(\mathbf{y}^{\text{s}}|\mathbf{p}^i)]\} + \dim \mathbf{p}^i \frac{\ln n_{\text{t}}}{n_{\text{t}}}.$$

Hypothesis testing can also be used. For a synoptic presentation of various available criteria, see, *e.g.*, (Söderström, 1977; Veres, 1991).

EXAMPLE 3.10

Assume that the data satisfy

$$\mathbf{y}(t_i) = \mathbf{y}_{\text{m}*}(t_i, \mathbf{p}^*) + \boldsymbol{\varepsilon}_i, \quad i = 1, \dots, n_{\text{t}},$$

where $\mathbf{y}_{\text{m}*}$ is the output of the model with the correct structure, for the true value $\mathbf{p}^*$ of its parameters, and the $\boldsymbol{\varepsilon}_i$'s are i.i.d. $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, with $\boldsymbol{\Sigma}$ unknown. The vectors $\mathbf{y}$, $\mathbf{y}_{\text{m}}$ and $\boldsymbol{\varepsilon}_i$ are n_{y}-dimensional. As in Example 3.4, the log-likelihood can then be written as

$$\ln \pi_{\text{y}}(\mathbf{y}^{\text{s}}|\mathbf{p}_{\text{e}}) = -\frac{n_{\text{y}}n_{\text{t}}}{2} \ln 2\pi - \frac{n_{\text{t}}}{2} \ln \det \boldsymbol{\Sigma}$$

$$-\frac{1}{2}\sum_{i=1}^{n_{\text{t}}} [\mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \mathbf{p})]^{\text{T}}\boldsymbol{\Sigma}^{-1}[\mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \mathbf{p})],$$

where $\mathbf{p}_{\text{e}}$ is an extended vector of unknown parameters consisting of $\mathbf{p}$ and the entries of the upper triangular part of $\boldsymbol{\Sigma}$. The AIC cost function can therefore be written as

$$j_{\text{aic}}(M, \mathbf{p}_{\text{e}}) = \frac{n_{\text{y}}}{2} \ln 2\pi + \frac{1}{2} \ln \det \boldsymbol{\Sigma}$$

$$+\frac{1}{2n_{\text{t}}}\sum_{i=1}^{n_{\text{t}}} [\mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \mathbf{p})]^{\text{T}}\boldsymbol{\Sigma}^{-1}[\mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \mathbf{p})] + \frac{1}{n_{\text{t}}} \dim \mathbf{p}_{\text{e}}.$$

Assume first that the model structure is fixed. The dimension of $\mathbf{p}_{\text{e}}$ is then constant, and $\widehat{\mathbf{p}}_{\text{e}}$ is given by the maximum-likelihood estimator $(\widehat{\mathbf{p}}_{\text{ml}}, \widehat{\boldsymbol{\Sigma}}_{\text{ml}})$, already obtained. The AIC cost function becomes

$$j_{\text{aic}}(M, \widehat{\mathbf{p}}_{\text{eml}}) = \frac{n_{\text{y}}}{2} \ln 2\pi + \frac{1}{2} \ln \det \widehat{\boldsymbol{\Sigma}}_{\text{ml}}$$

$$+\frac{1}{2n_{\text{t}}}\sum_{i=1}^{n_{\text{t}}} [\mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \widehat{\mathbf{p}}_{\text{ml}})]^{\text{T}}\, \widehat{\boldsymbol{\Sigma}}_{\text{ml}}^{-1}\, [\mathbf{y}(t_i) - \mathbf{y}_{\text{m}}(t_i, \widehat{\mathbf{p}}_{\text{ml}})] + \frac{1}{n_{\text{t}}} \dim \mathbf{p}_{\text{e}},$$

to be minimized with respect to the structure M. The third term on the right-hand side can be computed as

$$\frac{1}{2n_t}\sum_{i=1}^{n_t}[\mathbf{y}(t_i)-\mathbf{y}_m(t_i,\hat{\mathbf{p}}_{ml})]^T\hat{\boldsymbol{\Sigma}}_{ml}^{-1}[\mathbf{y}(t_i)-\mathbf{y}_m(t_i,\hat{\mathbf{p}}_{ml})]$$

$$=\frac{1}{2n_t}\text{trace}\left\{\sum_{i=1}^{n_t}[\mathbf{y}(t_i)-\mathbf{y}_m(t_i,\hat{\mathbf{p}}_{ml})][\mathbf{y}(t_i)-\mathbf{y}_m(t_i,\hat{\mathbf{p}}_{ml})]^T\hat{\boldsymbol{\Sigma}}_{ml}^{-1}\right\}$$

$$=\frac{1}{2}\text{trace}\,(\hat{\boldsymbol{\Sigma}}_{ml}\hat{\boldsymbol{\Sigma}}_{ml}^{-1})=\frac{1}{2}\text{trace}\,\mathbf{I}_{n_y}=\frac{n_y}{2}.$$

It is thus independent of the structure considered. During the structure selection phase, it therefore suffices to find M that minimizes

$$j_{aic}(M,\hat{\mathbf{p}}_{e_{ml}})=\frac{1}{2}\ln\det\hat{\boldsymbol{\Sigma}}_{ml}+\frac{1}{n_t}\dim\mathbf{p}_e,$$

where

$$\hat{\boldsymbol{\Sigma}}_{ml}=\frac{1}{n_t}\sum_{i=1}^{n_t}[\mathbf{y}(t_i)-\mathbf{y}_m(t_i,\hat{\mathbf{p}}_{ml})][\mathbf{y}(t_i)-\mathbf{y}_m(t_i,\hat{\mathbf{p}}_{ml})]^T. \qquad \lozenge$$

REMARKS 3.8

— As already noted in Example 3.4, $\hat{\mathbf{p}}_{ml}$ can be obtained in Example 3.10 by minimizing

$$j_{ml}(\mathbf{p}) = \ln\det\sum_{i=1}^{n_t}[\mathbf{y}(t_i)-\mathbf{y}_m(t_i,\mathbf{p})][\mathbf{y}(t_i)-\mathbf{y}_m(t_i,\mathbf{p})]^T,$$

independently of $\boldsymbol{\Sigma}$. The choice of M can therefore be made by minimizing

$$j_{aic}(M,\hat{\mathbf{p}}_{e_{ml}})=\frac{1}{2}j_{ml}(\hat{\mathbf{p}}_{ml})+\frac{1}{n_t}\dim\mathbf{p}_e.$$

It is therefore not necessary to consider $\boldsymbol{\Sigma}$ explicitly. Moreover, the number of unknown parameters in $\boldsymbol{\Sigma}$ does not depend on the structure, so dim $\mathbf{p}_e$ may be replaced by dim $\mathbf{p}$. When the output is scalar, the cost becomes

$$j_{aic}(M,\hat{\mathbf{p}}_{e_{ml}})=\frac{1}{2}\ln\sum_{i=1}^{n_t}[y(t_i)-y_m(t_i,\hat{\mathbf{p}}_{ml})]^2+\frac{1}{n_t}\dim\,\mathbf{p}.$$

— Criteria based on statistical considerations are not the only ones that can be employed to select the model structure to be used. An especially telling test is to compare the performances of the best models obtained for each structure on validation data not used to estimate the model parameters. Chapter 7 will present other methods that can be employed to eliminate model structures by revealing their defects.

— Data quality is an essential ingredient for the selection of a suitable model structure. Section 6.6.3 presents tools to design an optimal experiment to collect data to discriminate between competing model structures. ◊

3.5 Bayesian criteria

Maximum-likelihood estimation considers $\mathbf{p}$ as unknown but with a single actual value. Bayesian approaches instead consider a distribution of possible values for $\mathbf{p}$. Before the observations are made, $\mathbf{p}$ is assumed to have a known prior probability density $\pi_p(\mathbf{p})$. The joint probability density of $\mathbf{y}^s$ and $\mathbf{p}$ satisfies

$$\pi(\mathbf{y}^s, \mathbf{p}) = \pi_y(\mathbf{y}^s|\mathbf{p})\pi_p(\mathbf{p}) = \pi_p(\mathbf{p}|\mathbf{y}^s)\pi_y(\mathbf{y}^s).$$

The posterior probability density for $\mathbf{p}$ (taking the data $\mathbf{y}^s$ into account) is therefore given by

$$\pi_p(\mathbf{p}|\mathbf{y}^s) = \frac{\pi_y(\mathbf{y}^s|\mathbf{p})\pi_p(\mathbf{p})}{\pi_y(\mathbf{y}^s)} .$$

This is *Bayes' rule*, which gives its name to this class of estimators and quantifies what has been learnt by collecting data. Since $\mathbf{y}^s$ is a vector of known numbers, $\pi_y(\mathbf{y}^s)$ is just a normalization constant ensuring that $\pi_p(\mathbf{p}|\mathbf{y}^s)$ is a probability density,

$$\pi_y(\mathbf{y}^s) = \int_{\mathbb{P}} \pi_y(\mathbf{y}^s|\mathbf{p})\pi_p(\mathbf{p})d\mathbf{p}.$$

To compute the probability density of $\mathbf{p}$ conditional on the data $\mathbf{y}^s$, one therefore only need know how to express $\pi_y(\mathbf{y}^s|\mathbf{p})$ by taking advantage of the information or hypotheses on the noise (as was done in Section 3.3) and to have $\pi_p(\mathbf{p})$ at one's disposal, which expresses prior knowledge on the parameters. This knowledge may result from previous measurements on the same process or on similar processes (Section 3.3.4). The *maximum-entropy* approach (see, *e.g.*, (Mohammad-Djafari and Demoment, 1988, 1993)) makes it possible to choose a distribution $\pi_p(\mathbf{p})$ that is consistent with prior knowledge but does not introduce extraneous information. It suggests choosing the prior probability density π_p by maximizing the (Shannon) entropy

$$\mathcal{H}(\pi_p) = -\int_{\mathbb{P}} \pi_p(\mathbf{p}) \ln\,[\pi_p(\mathbf{p})]\; d\mathbf{p},$$

under the constraints expressing the available prior information. The optimal density is obtained by a Lagrange-multiplier technique. If, for instance, only the prior mean $\mathbf{p}_0$ and prior variance $\boldsymbol{\Omega}$ of $\mathbf{p}$ are known, the maximum-entropy principle leads to a Gaussian prior $\mathcal{N}(\mathbf{p}_0, \boldsymbol{\Omega})$. If the only prior information on the parameters is that

$$\mathbf{p}_{\min} \leq \mathbf{p} \leq \mathbf{p}_{\max},$$

the optimal prior density π_p will be uniform on the box thus defined. See also (Box and Tiao, 1973) for the notion of uninformative prior.

Taking advantage of the posterior probability density $\pi_p(\mathbf{p}|\mathbf{y}^s)$ is not always easy, and it is often desirable to obtain a point estimate of the parameters, *i.e.* a unique numerical value $\hat{\mathbf{p}}$. The next two sections present rules that may be used for this purpose.

REMARK 3.9

When the prior distribution is discrete, applying Bayes' rule becomes particularly simple, because only the weights (discrete probabilities) associated with the support points need be updated. The computation of expected values is also drastically simplified, since integrals are replaced by discrete sums. This is an additional advantage of the nonparametric methods for estimating the distribution of parameters in a population presented in Section 3.3.4, which lead to discrete distributions. ◊

3.5.1 Maximum *a posteriori*

Maximum *a posteriori* (or MAP) estimation searches for $\hat{\mathbf{p}}_{\mathrm{map}}$ that maximizes

$$j_{\mathrm{map}}(\mathbf{p}) = \pi_p(\mathbf{p}|\mathbf{y}^s) = \frac{\pi_y(\mathbf{y}^s|\mathbf{p})\pi_p(\mathbf{p})}{\pi_y(\mathbf{y}^s)} .$$

As $\pi_y(\mathbf{y}^s)$ does not depend on $\mathbf{p}$, this is equivalent to maximizing $\pi_y(\mathbf{y}^s|\mathbf{p})\pi_p(\mathbf{p})$, or

$$j_{\mathrm{map}}(\mathbf{p}) = \ln \pi_y(\mathbf{y}^s|\mathbf{p}) + \ln \pi_p(\mathbf{p}),$$

since logarithm is a monotonically increasing function. The first term of this sum is nothing but the log-likelihood, and the second one expresses the prior information on the parameters. It is thus easy to incorporate some (objective or subjective) information on the possible values for $\mathbf{p}$.

Under the hypotheses H1-H5 (or H1-H5') of Section 3.3.3 and the additional condition

H6: $\pi_p(\mathbf{p})$ is continuous and nonzero in a neighbourhood of $\mathbf{p}^*$,

the MAP estimator shares the asymptotic consistency and efficiency properties of the maximum-likelihood estimator. It is also invariant under reparametrization. An approximation for its non-asymptotic density when the measurement noise is Gaussian can be found in Section 6.4.1.

EXAMPLE 3.11

Assume that the prior distribution of $\mathbf{p}$ is uniform on the box $\mathbb{P}$ defined by

$$p_{i_{\min}} \leq p_i \leq p_{i_{\max}},\ i = 1, \dots, n_p.$$

For any $\mathbf{p}$ that does not satisfy these inequalities, $\pi_p(\mathbf{p}) = 0$ and $\ln \pi_p(\mathbf{p}) = -\infty$. One is thus sure that $\hat{\mathbf{p}}_{\mathrm{map}}$ will satisfy the constraints that define $\mathbb{P}$. For any $\mathbf{p}$ in $\mathbb{P}$, $\pi_p(\mathbf{p})$ and $\ln \pi_p(\mathbf{p})$ have a constant finite value. Hence, if $\hat{\mathbf{p}}_{\mathrm{ml}}$ belongs to $\mathbb{P}$, $\hat{\mathbf{p}}_{\mathrm{map}} = \hat{\mathbf{p}}_{\mathrm{ml}}$. ◊

REMARK 3.10

This example does not provide any practical method to impose inequality constraints upon $\mathbf{p}$. Using the MAP criterion for that purpose would introduce discontinuities that would strongly complicate its optimization. Methods to impose constraints upon $\mathbf{p}$ will be presented in Section 3.6. ◊

EXAMPLE 3.12

Assume that the prior distribution for the parameters is $\mathcal{N}(\mathbf{p}_0, \boldsymbol{\Omega})$, with $\mathbf{p}_0$ and $\boldsymbol{\Omega}$ known. The prior mean $\mathbf{p}_0$ may for example be a maximum-likelihood estimate of $\mathbf{p}$ obtained from preliminary measurements, and the prior covariance $\boldsymbol{\Omega}$, which characterizes the uncertainty in $\mathbf{p}_0$, may have been obtained by one of the methods described in Chapter 5.

The prior probability density for the parameters satisfies

$$\pi_{\mathrm{p}}(\mathbf{p}) = [(2\pi)^{n_{\mathrm{p}}} \det \boldsymbol{\Omega}]^{-1/2} \exp\,[-\frac{1}{2}(\mathbf{p}-\mathbf{p}_0)^{\mathrm{T}}\boldsymbol{\Omega}^{-1}(\mathbf{p}-\mathbf{p}_0)].$$

Up to a constant term, the MAP cost function can therefore be written as

$$j_{\mathrm{map}}(\mathbf{p}) = \ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}) - \frac{1}{2}(\mathbf{p}-\mathbf{p}_0)^{\mathrm{T}}\boldsymbol{\Omega}^{-1}(\mathbf{p}-\mathbf{p}_0).$$

Prior information is here incorporated by subtracting a quadratic penalty to the log-likelihood. *Regularization techniques*, used, *e.g.*, in image processing, also lead to quadratic penalty functions, which can be given a Bayesian interpretation (Demoment, 1989). When

$$\mathbf{y}^{\mathrm{s}} = \mathbf{R}\mathbf{p}^{*} + \mathbf{n},$$

where $\mathbf{n}$ is distributed $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ (with $\boldsymbol{\Sigma}$ known), $\hat{\mathbf{p}}_{\mathrm{map}}$ is given by

$$\hat{\mathbf{p}}_{\mathrm{map}} = (\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R} + \boldsymbol{\Omega}^{-1})^{-1}(\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{y}^{\mathrm{s}} + \boldsymbol{\Omega}^{-1}\mathbf{p}_0). \qquad ◊$$

A feature of MAP estimators is that the number of data points need not exceed the number of parameters in order to ensure uniqueness of the estimate. Thus, for instance, the number of measurements can be limited in clinical medicine for cost and patient comfort, provided that reliable information is available on the possible values of the parameters in the population which the patient belongs to.

3.5.2 Minimum risk

Assume that one can evaluate the *cost* $j(\mathbf{p}|\mathbf{p}^{*})$ of believing that the value of the parameters is $\mathbf{p}$ when it is actually $\mathbf{p}^{*}$. Since $\mathbf{p}^{*}$ is unknown, one can search for $\hat{\mathbf{p}}_{\mathrm{mr}}$ that minimizes the *risk*, defined as the mean of this cost over all possible values of $\mathbf{p}^{*}$, conditioned on the observations $\mathbf{y}^{\mathrm{s}}$. Provided that the integral exists, the risk is then

$$j_{\mathrm{mr}}(\mathbf{p}) = \int_{\mathbb{P}} j(\mathbf{p}|\mathbf{p}^*)\pi_{\mathrm{p}}(\mathbf{p}^*|\mathbf{y}^{\mathrm{s}})\mathrm{d}\mathbf{p}^*.$$

This risk must now be expressed as a function of the quantities that are assumed to be available. From Bayes' rule

$$j_{\mathrm{mr}}(\mathbf{p}) = \frac{1}{\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}})} \int_{\mathbb{P}} j(\mathbf{p}|\mathbf{p}^*)\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*)\pi_{\mathrm{p}}(\mathbf{p}^*)\mathrm{d}\mathbf{p}^*.$$

Since $\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}})$ does not depend on $\mathbf{p}$, one can equivalently minimize

$$j_{\mathrm{mr}}(\mathbf{p}) = \int_{\mathbb{P}} j(\mathbf{p}|\mathbf{p}^*)\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*)\pi_{\mathrm{p}}(\mathbf{p}^*)\mathrm{d}\mathbf{p}^*.$$

In this expression, $j(\mathbf{p}|\mathbf{p}^*)$ expresses what is known about the aim of the modelling, $\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*)$ expresses the information on the noise acting on the system and $\pi_{\mathrm{p}}(\mathbf{p}^*)$ expresses prior information on the parameters.

REMARKS 3.11

— Minimum-risk estimation may lead to very complex computation, which are drastically simplified if the prior distribution of the parameters is discrete, for the computation of $j_{\mathrm{mr}}(\mathbf{p})$ reduces to a discrete summation. Such will be the case, for example, if the prior distribution has been obtained by nonparametric maximum-likelihood estimation of the distribution of the parameters in a population (Section 3.3.4).

— When the prior distribution is not discrete, evaluating $j_{\mathrm{mr}}(\mathbf{p})$ requires multiple integration (see, *e.g.*, (Genz and Malik, 1980) for an integration algorithm), which raises critical numerical problems. Stochastic approximation (Sections 4.3.8 and 6.4.3.2), however, makes it possible to compute $\hat{\mathbf{p}}_{\mathrm{mr}}$ without ever evaluating $j_{\mathrm{mr}}(\mathbf{p})$.

— $\hat{\mathbf{p}}_{\mathrm{mr}}$ may be very sensitive to the tails of the distribution $\pi_{\mathrm{p}}(\mathbf{p})$ (Bard, 1974).

— Contrary to maximum-likelihood and MAP estimators, $\hat{\mathbf{p}}_{\mathrm{mr}}$ is not invariant to reparametrization. ◊

EXAMPLE 3.13

Assume that $j(\mathbf{p}|\mathbf{p}^*)$ is quadratic in $\mathbf{p}$

$$j(\mathbf{p}|\mathbf{p}^*) = (\mathbf{p} - \mathbf{p}^*)^{\mathrm{T}}\mathbf{Q}(\mathbf{p} - \mathbf{p}^*),$$

with $\mathbf{Q}$ a symmetric positive-definite weighting matrix. The risk is then

$$j_{\mathrm{mr}}(\mathbf{p}) = \int_{\mathbb{P}} (\mathbf{p} - \mathbf{p}^*)^{\mathrm{T}}\mathbf{Q}(\mathbf{p} - \mathbf{p}^*)\pi_{\mathrm{p}}(\mathbf{p}^*|\mathbf{y}^{\mathrm{s}})\mathrm{d}\mathbf{p}^*.$$

The minimum-risk estimator must therefore satisfy

$$\frac{\partial}{\partial \mathbf{p}} j_{\mathrm{mr}}|_{\hat{\mathbf{p}}_{\mathrm{mr}}} = 0 = 2\mathbf{Q} \int_{\mathbb{P}} (\hat{\mathbf{p}}_{\mathrm{mr}} - \mathbf{p})\pi_{\mathrm{p}}(\mathbf{p}|\mathbf{y}^{\mathrm{s}})\mathrm{d}\mathbf{p},$$

or, since $\mathbf{Q}$ is invertible,

$$\hat{\mathbf{p}}_{\mathrm{mr}} \int_{\mathbb{P}} \pi_{\mathrm{p}}(\mathbf{p}|\mathbf{y}^{\mathrm{s}})\mathrm{d}\mathbf{p} = \hat{\mathbf{p}}_{\mathrm{mr}} = \int_{\mathbb{P}} \mathbf{p}\pi_{\mathrm{p}}(\mathbf{p}|\mathbf{y}^{\mathrm{s}})\mathrm{d}\mathbf{p} = \mathrm{E}\{\mathbf{p}|\mathbf{y}^{\mathrm{s}}\}.$$

Thus $\hat{\mathbf{p}}_{\mathrm{mr}}$ is the posterior mean of $\mathbf{p}$ *whatever* the non-singular weighting matrix $\mathbf{Q}$. From Bayes' rule, it is given by

$$\hat{\mathbf{p}}_{\mathrm{mr}} = \int_{\mathbb{P}} \mathbf{p} \frac{\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p})\pi_{\mathrm{p}}(\mathbf{p})}{\pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}})} \mathrm{d}\mathbf{p}.$$

The computation of $\hat{\mathbf{p}}_{\mathrm{mr}}$ therefore requires no optimization, unlike that of most estimates. It can nevertheless be performed analytically in simple cases only. Assume, for instance, that

$$\mathbf{y}^{\mathrm{s}} = \mathbf{R}\mathbf{p}^{*} + \mathbf{n},$$

where $\mathbf{n}$ is distributed $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ (with $\mathbf{\Sigma}$ known) and $\mathbf{p}$ is *a priori* distributed $\mathcal{N}(\mathbf{p}_0, \mathbf{\Omega})$ (with $\mathbf{p}_0$ and $\mathbf{\Omega}$ known). The estimator $\hat{\mathbf{p}}_{\mathrm{mr}}$ then coincides with the MAP estimator (Example 3.12). ◊

EXAMPLE 3.14

Assume that the aim of the modelling is optimal control of a process, in the sense of minimizing the cost function

$$j_{\mathrm{control}}(\mathbf{u}, \mathbf{p}) = \sum_{i=1}^{n_{\mathrm{i}}} \mathbf{x}_i^{\mathrm{T}}\mathbf{Q}_1\mathbf{x}_i + \mathbf{u}_i^{\mathrm{T}}\mathbf{Q}_2\mathbf{u}_i,$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are predefined weighting matrices.

To minimize this cost with respect to the sequence $\mathbf{u}$ of controls $\mathbf{u}_i$, one needs to know the parameters $\mathbf{p}$ of the model so as to predict the successive states $\mathbf{x}_i$ ($i = 1, \dots, n_{\mathrm{i}}$) of the system. If the optimal control sequence $\mathbf{u}_{\mathrm{opt}}(\mathbf{p})$ computed for a model with parameter values $\mathbf{p}$ is applied to a system with parameter values $\mathbf{p}^{*}$, the resulting deterioration of the control cost can be used as the identification cost

$$j(\mathbf{p}|\mathbf{p}^{*}) = j_{\mathrm{control}}(\mathbf{u}_{\mathrm{opt}}(\mathbf{p}), \mathbf{p}^{*}) - j_{\mathrm{control}}(\mathbf{u}_{\mathrm{opt}}(\mathbf{p}^{*}), \mathbf{p}^{*}).$$ ◊

3.6 Constraints on parameters

If the prior feasible space $\mathbb{P}$ for the parameters is not all of $\mathbb{R}^{n_p}$, it may become necessary to force the vector of estimates to stay within $\mathbb{P}$. Many approaches are available, and we shall only consider here those that transform the problem into an unconstrained one by modifying the cost function. Their advantage is compatibility with the unconstrained optimization techniques that form the core of Chapter 4. Constrained optimization will be considered in Section 4.3.4.

Note that when there are equality constraints (or active inequality constraints) restricting the parameters to a subset of parameter space, the Fisher information matrix used to characterize the uncertainty in the parameters (Section 3.3.3 and Chapter 5) should be replaced by a rank-reduced Fisher information matrix (Gorman and Hero, 1990).

3.6.1 Equality constraints

Equality constraints can be written as

$$\mathbf{c}_e(\mathbf{p}) = \mathbf{0},$$

where $\mathbf{c}_e$ may be a vector. They force the parameters to belong to a hypersurface of $\mathbb{R}^{n_p}$ and express that they are dependent. Whenever possible, one should then reparametrize the problem by expressing some parameters as functions of others so as to make all remaining parameters independent. This usually simplifies optimization noticeably by removing constraints and decreasing the dimension of parameter space to be explored.

When this approach is unfeasible, one can add a *penalty function* to the initial cost function $j(\mathbf{p})$ (assumed to be minimized). This penalty function will be zero as long as $\mathbf{c}_e(\mathbf{p}) = \mathbf{0}$, but will increase with violation of the constraints. One may, for instance, minimize

$$j_\mu(\mathbf{p}) = j(\mathbf{p}) + \frac{1}{2}\,\mu\,\|\mathbf{c}_e(\mathbf{p})\|_2^2,$$

where $\mu > 0$. If j and $\mathbf{c}_e$ are continuous in $\mathbf{p}$, any unconstrained global minimizer $\check{\mathbf{p}}_\mu$ of j_μ will converge to a global minimizer of j under the constraints $\mathbf{c}_e$ as μ tends to infinity (Polyak, 1987). Theoretically, convergence is thus achieved under very general conditions. In practice, however, the larger μ is the more ill-conditioned the unconstrained optimization problem becomes. Under some conditions (Bonnans, 1987; Hiriart-Urruty and Lemaréchal, 1993), a so-called *exact penalty technique*

$$j_\mu(\mathbf{p}) = j(\mathbf{p}) + \mu\,\|\mathbf{c}_e(\mathbf{p})\|_2$$

can be employed instead, which makes the constrained minimization of j possible by unconstrained minimization of j_μ, provided that μ is large enough (but finite).

Augmented Lagrangian techniques also permit the requirement that μ should tend to infinity to be dropped (Minoux, 1983; Polyak, 1987). The Lagrangian of the initial optimization problem can be written as

$$\mathcal{L}(\mathbf{p}, \mathbf{d}) = j(\mathbf{p}) + \mathbf{d}^{\mathrm{T}}\mathbf{c}_{\mathrm{e}}(\mathbf{p}),$$

where $\mathbf{d}$ is the dual vector of $\mathbf{p}$. The Lagrangian formulation allows the elimination of the constraints in the study of the theoretical optimality conditions. At the optimum,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}} = \mathbf{0}, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{d}} = \mathbf{0},$$

and this point corresponds to a minimum with respect to $\mathbf{p}$ and a maximum with respect to $\mathbf{d}$. $\mathcal{L}(\mathbf{p}, \mathbf{d})$ should therefore be minimized with respect to $\mathbf{p}$ and maximized with respect to $\mathbf{d}$. More rapidly convergent methods use an augmented Lagrangian that is the sum of the Lagrangian and a penalty function

$$\mathcal{L}_{\mathrm{a}}(\mathbf{p}, \mathbf{d}, \mu) = j(\mathbf{p}) + \mathbf{d}^{\mathrm{T}}\mathbf{c}_{\mathrm{e}}(\mathbf{p}) + \frac{1}{2}\mu \,\|\mathbf{c}_{\mathrm{e}}(\mathbf{p})\|_2^2.$$

A possible technique consists at iteration i of performing one unconstrained minimization with respect to $\mathbf{p}$

$$\mathbf{p}^i = \arg \min_{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}}} \mathcal{L}_{\mathrm{a}}(\mathbf{p}, \mathbf{d}^{i-1}, \mu),$$

followed by *one step* of maximization with respect to $\mathbf{d}$ by the gradient method (Section 4.3.3.1)

$$\mathbf{d}^i = \mathbf{d}^{i-1} + \mu\mathbf{c}_{\mathrm{e}}(\mathbf{p}^i).$$

If the second derivatives of the cost function j and constraints $\mathbf{c}_{\mathrm{c}}$ are Lipschitzian and if μ is large enough (but finite), then convergence will occur whatever the initial value of $\mathbf{d}$. In principle, the larger μ is the faster the convergence should be, but in practice ill-conditioning limits the value that can be given to μ.

3.6.2 Inequality constraints

Inequality constraints will be written as

$$\mathbf{c}_{\mathrm{i}}(\mathbf{p}) \leq \mathbf{0},$$

where the inequality is to be taken componentwise. The approaches to transform the problem into unconstrained optimization are similar to those for equality constraints. One may for example minimize

$$j_\mu(\mathbf{p}) = j(\mathbf{p}) + \frac{1}{2}\mu \,\|\mathbf{c}_{\mathrm{i}}(\mathbf{p})_+\|_2^2,$$

where the entries of $\mathbf{c}_{\mathrm{i}}(\mathbf{p})_+$ are equal to the positive parts of those of $\mathbf{c}_{\mathrm{i}}(\mathbf{p})$. Such a penalty function is said to be *exterior*, because it only acts when the constraints are violated. Here too, the theoretical requirement that μ should tend to infinity results in practical problems of ill conditioning.

Such problems can be avoided by using an augmented Lagrangian and the algorithm mentioned for equality constraints. For more details on the restrictions and advantages of this very powerful method, see (Polyak, 1987). Exact penalty functions

$$j_\mu(\mathbf{p}) = j(\mathbf{p}) + \mu \, \|\mathbf{c}_i(\mathbf{p})_+\|_2,$$

or *interior penalty functions*, which act before the constraints are violated, may also be used.

Reparametrization techniques can also be employed. To impose for example $p_i \geq 0$, one may replace p_i by $(\exp q_i)$ with $q_i \in]-\infty, +\infty[$, which guarantees the inviolability of the constraint. Similarly, replacing p_i by $(\tanh q_i)(b - a)/2 + (a + b)/2$, with $q_i \in]-\infty, +\infty[$, guarantees that $a < p_i < b$.

REMARK 3.12

Incorporating the inequality constraints defining the prior feasible space for the parameters into the optimization problem may be inadvisable. In the context of *parameter estimation*, it excludes any better optimizer that might exist outside $\mathbb{P}$ and might lead one to reconsider one's model structure, data or prior feasible space. It therefore seems reasonable to try unconstrained optimization first. If the resulting optimizer $\hat{\mathbf{p}}$ is in $\mathbb{P}$, the problem is solved. Otherwise, constrained optimization may be tried, but often yields an optimizer $\hat{\mathbf{p}}$ on at least one of the constraints. The associated model $M(\hat{\mathbf{p}})$ is at the boundary of what is acceptable, so the structure M is probably unsuitable. The constrained optimizer $\hat{\mathbf{p}}$ is more easily accepted if no constraint is active at $\hat{\mathbf{p}}$ (Figure 3.8). Conversely, in *optimal control* or *experiment design* (Chapter 6), it is essential to take inequality constraints into account, because usually some of them are active at the optimum. ◊

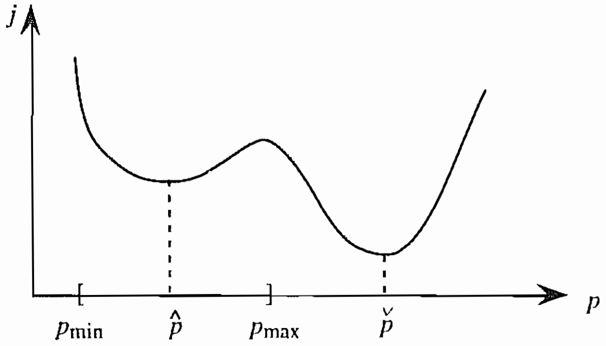


Figure 3.8. Constraints may lead to a local minimizer $\hat{p}$ strictly inside the feasible domain whereas the global minimizer $\check{p}$ is unfeasible

3.7 Robustness

An estimator is said to be robust if its performance does not deteriorate too much when the hypotheses on which it is based are not satisfied. One may, for instance, have assumed that the noise was $\mathcal{N}(0, \sigma^2)$ when its actual probability density has heavier tails than a Gaussian distribution. The data may also have been contaminated by outliers resulting from errors in data collection or faulty sensors.

Some of the estimators considered so far show very little robustness. The definition of robust estimators has given rise to many theoretical contributions (Launer and Wilkinson, 1979; Huber, 1981; Rousseeuw and Leroy, 1987), but we shall only mention some practical tools.

3.7.1 Robustness to uncertainty on the noise distribution

We have seen that if the observations satisfy

$$y(t_i) = y_{\mathrm{m}}(t_i, \mathbf{p}^*) + \varepsilon(t_i), \quad i = 1, \dots, n_{\mathrm{t}},$$

where the sequence of $\varepsilon(t_i)$ is i.i.d. with the probability density function π_ε, the log-likelihood function can be written as

$$\ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}) = \sum_{i=1}^{n_{\mathrm{t}}} \ln \pi_\varepsilon[y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})].$$

When π_ε is reliably known, maximum-likelihood estimation asymptotically achieves

$$\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}\{(\hat{\mathbf{p}}_{\mathrm{ml}} - \mathbf{p}^*)(\hat{\mathbf{p}}_{\mathrm{ml}} - \mathbf{p}^*)^{\mathrm{T}}\} = \mathbf{F}^{-1}(\mathbf{p}^*),$$

as the number of data points tends to infinity. The Fisher information matrix can be written as

$$\mathbf{F}(\mathbf{p}^*, \pi_\varepsilon) = I(\pi_\varepsilon) \sum_{i=1}^{n_{\mathrm{t}}} \frac{\partial}{\partial \mathbf{p}} [y_{\mathrm{m}}(t_i, \mathbf{p})]\Big|_{\mathbf{p}^*} \frac{\partial}{\partial \mathbf{p}^{\mathrm{T}}} [y_{\mathrm{m}}(t_i, \mathbf{p})]\Big|_{\mathbf{p}^*},$$

where the scalar $I(\pi_\varepsilon)$ is the *Fisher information*

$$I(\pi_\varepsilon) = \int_{\mathbb{D}} \frac{1}{\pi_\varepsilon(\varepsilon)} \left(\frac{\mathrm{d}\pi_\varepsilon(\varepsilon)}{\mathrm{d}\varepsilon}\right)^2 \mathrm{d}\varepsilon,$$

with $\mathbb{D} = \{\varepsilon \mid \pi_\varepsilon(\varepsilon) > 0\}$. It must, of course, be assumed that this integral exists. Information matrices computed for various densities π_ε only differ in $I(\pi_\varepsilon)$. For instance, for $\mathcal{N}(0, \sigma^2)$, $I(\pi_\varepsilon) = 1/\sigma^2$. Asymptotically, the uncertainty in the parameters will increase if the Fisher information decreases.

Assume now that π_ε is only known to belong to some family $\mathbb{F}$. A minimax approach may then be adopted (Huber, 1981; Polyak and Tsypkin, 1980; Tsypkin, 1983). It defines the robust estimator $\hat{\mathbf{p}}_\mathrm{r}$ as the maximum-likelihood estimator associated with the worst possible noise distribution $\hat{\pi}_\varepsilon$ in the sense of minimum Fisher information:

$$\hat{\pi}_\varepsilon = \arg \min_{\pi_\varepsilon \in \mathbb{F}} I(\pi_\varepsilon),$$

i.e. as the best estimator in the worst case. Under some regularity conditions, $\hat{\mathbf{p}}_\mathrm{r}$ is consistent, and the asymptotic covariance of the estimation error satisfies

$$\mathrm{E}[(\hat{\mathbf{p}}_\mathrm{r} - \mathbf{p}^*)(\hat{\mathbf{p}}_\mathrm{r} - \mathbf{p}^*)^\mathrm{T}] \le \mathbf{F}^{-1}(\mathbf{p}^*, \hat{\pi}_\varepsilon).$$

Equality is reached when $\pi_\varepsilon = \hat{\pi}_\varepsilon$, the worst case.

EXAMPLE 3.15

If $\mathbb{F}$ is the family of distributions for which the Fisher information is defined and $\pi_\varepsilon(0) \ge 1/(2a) > 0$, then $\hat{\pi}_\varepsilon$ is the Laplace distribution with zero mean and standard deviation $a\sqrt{2}$. The robust estimator obtained by this minimax approach is therefore the (unweighted) least-modulus estimator

$$\hat{\mathbf{p}}_\mathrm{r} = \arg \min \sum_{i=1}^{n_\mathrm{t}} |y(t_i) - y_\mathrm{m}(t_i, \mathbf{p})|.$$

Note that it is not necessary to know the value of a.

This very simple estimator also proves to be much more robust to outliers than least squares, as evidenced, *e.g.*, in (Venot *et al.*, 1986). Two hundred data points were generated according to

$$y(t) = 1000 \exp(-0.012t) + \varepsilon(t), \quad t = 1, \ldots, \quad 200,$$

with $\varepsilon(t)$ belonging to a sequence of i.i.d. $\mathcal{N}(0, 20)$ random variables. Thirty five outliers were then introduced, replacing some data points by zero, namely $y(t) = 0$, $t \in [10, 29] \cup [50, 64]$. The resulting data set was used to estimate the parameters of the model $y_\mathrm{m}(t, \mathbf{p}) = p_1 \exp(-p_2 t)$. The unweighted least-modulus estimate is $\hat{\mathbf{p}}_\mathrm{r} = (964, 0.0117)^\mathrm{T}$, reasonably close to $\mathbf{p}^*$ compared to the unweighted least-squares estimate $\hat{\mathbf{p}}_\mathrm{ls} = (450, 0.0063)^\mathrm{T}$. ◊

EXAMPLE 3.16

If $\mathbb{F}$ is the family of distributions for which the Fisher information is defined and the variance is finite,

$$\int_{-\infty}^{+\infty} z^2 \pi_\varepsilon(z)\mathrm{d}z \leq \sigma^2,$$

then the worst distribution is Gaussian, with zero mean and constant variance, which suggests using unweighted least squares. ◊

REMARK 3.13

This approach to robustness still assumes that the noise can be described as a sequence of i.i.d. random variables. Possible correlations and nonstationarities are therefore not explicitly taken into account. ◊

3.7.2 Breakdown point

A point estimator is a rule that associates an estimated value $\hat{\mathbf{p}}(\mathbf{y}^\mathrm{s})$ with any given data set $\mathbf{y}^\mathrm{s}$. Assume that the prior feasible domain for the parameters is $\mathbb{R}^{n_\mathrm{p}}$. Let $\mathbf{y}^\mathrm{o}$ be the data set obtained by replacing a percentage α of the data points in $\mathbf{y}^\mathrm{s}$ by outliers. The maximum bias of the estimator that can be induced by these outliers can be characterized by

$$\mathrm{bias}(\alpha, \hat{\mathbf{p}}(.), \mathbf{y}^\mathrm{s}) = \max_{\mathbf{y}^\mathrm{o}} \|\hat{\mathbf{p}}(\mathbf{y}^\mathrm{s}) - \hat{\mathbf{p}}(\mathbf{y}^\mathrm{o})\|.$$

The *breakdown point* of the estimator $\hat{\mathbf{p}}(.)$ is the smallest value of α such that $\mathrm{bias}(\alpha, \hat{\mathbf{p}}(.), \mathbf{y}^\mathrm{s}) = \infty$. For LP model structures, theoretical results can be established (Rousseeuw and Leroy, 1987). A single outlier tending to infinity is enough to break down least-squares, least-modulus and minimax estimators. Their breakdown points therefore tend to zero as the number of data points tends to infinity. Now, under suitable experimental conditions, estimators exist with a breakdown point that tends to 50% as the number of data points tends to infinity. Such estimators are thus much more robust to totally aberrant data points. This is the best achievable result. No reasonable estimator (see (Rousseeuw and Leroy, 1987) for more details) can have a larger breakdown point than 50%, because one could always arrange a majority of outliers in such a way that they could be described by a model with the structure chosen, and these outliers would then be preferred to the minority of regular data points.

One may, for example, employ the *least-median-of-squares* estimator

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \operatorname*{med}_{i}\, e^2(t_i, \mathbf{p}),$$

where the residual $e(t_i, \mathbf{p})$ is for instance an output error

$$e(t_i, \mathbf{p}) = y(t_i) - y_\mathrm{m}(t_i, \mathbf{p}).$$

This estimator systematically rejects the 50% data points that correspond to the largest residuals (Figure 3.9), which makes it very robust to severe outliers, at the cost of possibly rejecting significant data. Another possibility, with better asymptotic efficiency, is to use the *least-trimmed-squares* estimator

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{i \in \mathbb{I}} e^2(t_i, \mathbf{p}),$$

where $\mathbb{I}$ is the set of all indexes i such that $e^2(t_i, \mathbf{p})$ is smaller than or equal to the median of the squares of all residuals.

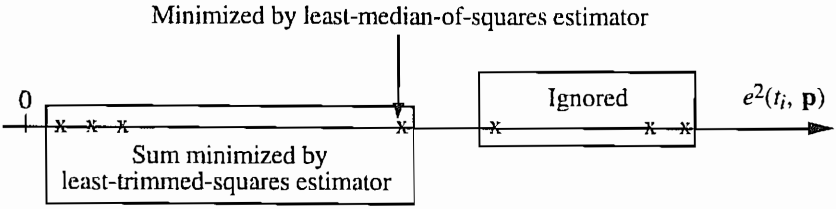

Minimized by least-median-of-squares estimator
0
Ignored
$e^2(t_i, \mathbf{p})$
Sum minimized by least-trimmed-squares estimator

Figure 3.9. Least-median-of-squares and least-trimmed-squares estimators

One drawback of the least-median-of-squares and least-trimmed-squares estimators is that they may reject data points in such a way that a large part of the response of the system is entirely ignored, which may result in a loss of identifiability. A bounded-error approach will be presented in Section 5.4.2.2 that escapes this problem while keeping a breakdown point that tends to 50%. This approach can even treat data sets with a majority of outliers, provided that these outliers are not describable by a model with the structure chosen.

REMARK 3.14

When it is not known whether outliers are present, one could compare the results obtained by least trimmed squares and least squares. If they turn out to be close, the least-squares estimate is probably more accurate, as it is based on twice as many samples. ◊

3.7.3 M-Estimators

An *M-estimator* (Huber, 1981) $\hat{\mathbf{p}}_{\mathrm{m}}$ minimizes the cost

$$j_{\mathrm{m}}(\mathbf{p}) = \sum_{i=1}^{n_t} \rho[e(t_i, \mathbf{p})].$$

The least-squares and least-modulus estimators are M-estimators, with $\rho(e) = e^2$ and $\rho(e) = |e|$ respectively. Many other functions with a minimum at $e = 0$ can be considered. One may thus use Huber's cost function (Figure 3.10):

$$\rho(e) = \begin{cases} \frac{1}{2} e^2 & \text{if } |e| \leq \delta, \\ \delta\,|e| - \frac{1}{2}\delta^2 & \text{otherwise,} \end{cases}$$

where δ is a threshold to be chosen. The resulting M-estimator therefore behaves as a least-squares estimator for small errors and as a least-modulus estimator when the errors get larger. Ljung (1987) suggests $\delta = \lambda\hat{\sigma}$, with $1 \leq \lambda \leq 1.8$ and $\hat{\sigma}$ an estimate of the standard deviation of the error that is robust to outliers:

$$\hat{\sigma} = \frac{\operatorname*{med}_i \{|e(t_i, \mathbf{p}) - \operatorname*{med}_i [e(t_i, \mathbf{p})]|\}}{0.7}.$$

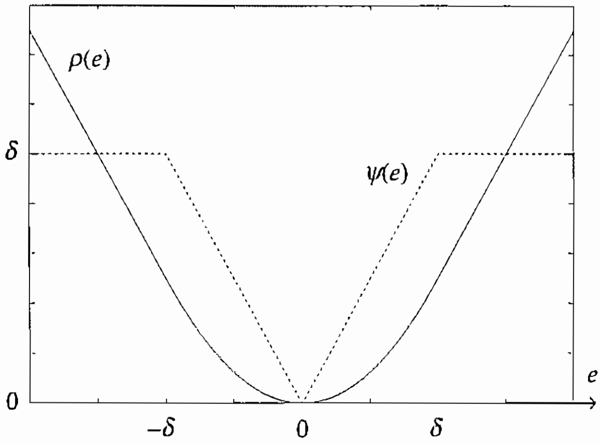


Figure 3.10. Huber's ρ and ψ (non-redescending M-estimator)

One may also use Tukey's cost function (Figure 3.11):

$$\rho(e) = \begin{cases} \frac{1}{2}\left(e^2 - \frac{e^4}{\delta^2} + \frac{e^6}{3\delta^4}\right) & \text{if } |e| \leq \delta, \\ \frac{\delta^2}{6} & \text{otherwise.} \end{cases}$$

REMARK 3.15

In the absence of constraints on $\mathbf{p}$, the optimizer $\hat{\mathbf{p}}_{\mathrm{m}}$ satisfies the stationarity conditions

$$\frac{\partial j_{\mathrm{m}}}{\partial \mathbf{p}}\Big|_{\hat{\mathbf{p}}_{\mathrm{m}}} = \mathbf{0}.$$

If $\psi = \frac{\partial \rho}{\partial e}$, these conditions become the *normal equations*

$$\sum_{i=1}^{n_t} \psi[e(t_i, \hat{\mathbf{p}}_{\mathrm{m}})] \frac{\partial e(t_i, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\hat{\mathbf{p}}_{\mathrm{m}}} = \mathbf{0}.$$

An M-estimator may be defined by specifying ψ or ρ; see Figures 3.10 and 3.11. The shape of ψ explains why Huber's M-estimator is said to be *non-redescending* whereas Tukey's is said to be *redescending*. ◊

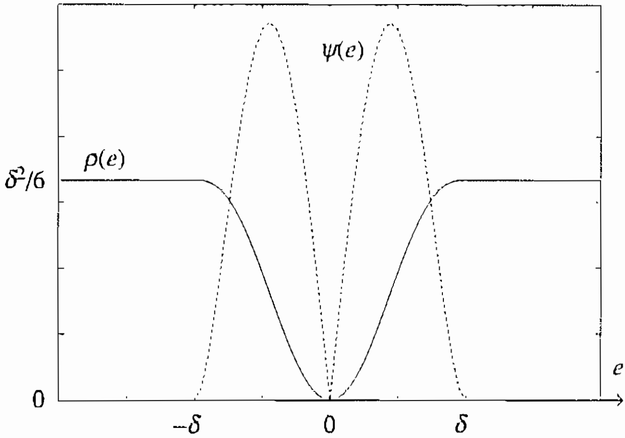


Figure 3.11. Tukey's ρ and ψ (redescending M-estimator)

3.7.4 Image processing example

There are many fields (medicine, satellite remote sensing, industrial inspection, astronomy...) where significant differences between images have to be detected. Figure 3.12 provides an example of two such images.

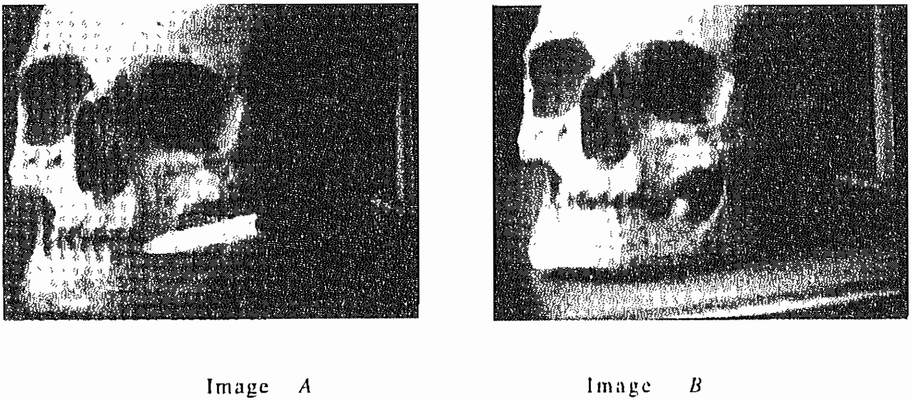


Figure 3.12. Images to be compared

In addition to significant differences, the two images usually have insignificant changes, due for instance to movement of the subject or retuning of the imaging device.

Pixel-by-pixel computation of the difference between the two images then often yields a useless image. This is why a calibration step is required, during which one of the images (say, Image B) is transformed with respect to a vector $\mathbf{p}$ of parameters, *e.g.*,

$$\mathbf{p} = \begin{bmatrix} \text{rotation} \\ x \text{ translation} \\ y \text{ translation} \\ \text{contrast} \\ \text{brightness} \\ \text{enlargement} \end{bmatrix}.$$

Let $B(\mathbf{p})$ be Image B after the transformation. The vector $\mathbf{p}$ is estimated so as to make $B(\mathbf{p})$ fit the reference image A best as measured by a cost function $j(\mathbf{p})$. The resulting scheme, Figure 3.13, is similar to Figure 1.5. The situation is rather unusual, because the significant differences play the role of outliers that hinder estimation of $\mathbf{p}$ (which is of no interest, but has to be used). The presence of outliers is therefore unavoidable, so a robust estimator is needed (Herbin *et al.*, 1989), such as the least-median-of-squares or least-trimmed-squares estimator or the outlier minimal number estimator presented in Section 5.4.2.2. Another possible estimator (Venot *et al.*, 1986) maximizes the number of sign changes in the image $A - B(\mathbf{p})$ scanned line by line. Such scanning transforms images into one-dimensional signals. The larger the number of sign changes is, the closer the signals associated with A and $B(\mathbf{p})$ are. Robustness comes from the fact that only the sign of the error is taken into consideration, not its magnitude. Very large errors corresponding to significant differences therefore have no more influence than the others.

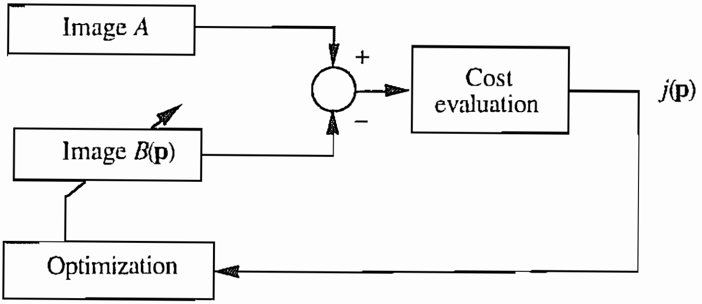


Figure 3.13. Principle of robust calibration

Figure 3.14 presents the image $B(\hat{\mathbf{p}})$ thus obtained, and Figure 3.15 the pixel-by-pixel difference of Images A and $B(\hat{\mathbf{p}})$. As can be seen, the detection of significant differences is now much easier! This technique sometimes makes it possible to reveal differences which could not be detected by direct inspection of the original images. A study of the properties of this type of estimator can be found in (Walter, Pronzato and Venot, 1989).

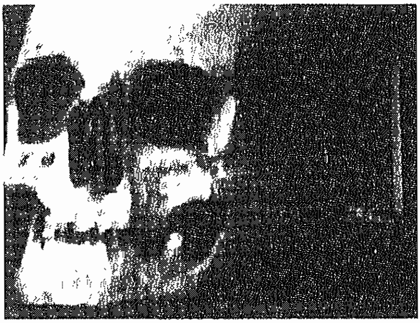


Figure 3.14. Image $B(\hat{\mathbf{p}})$ after robust calibration

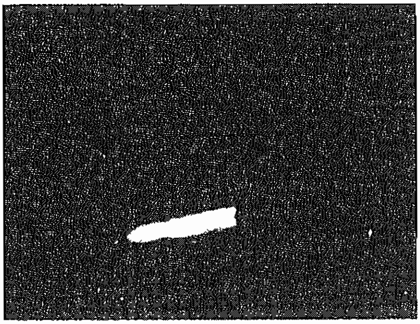


Figure 3.15. Difference $A - B(\hat{\mathbf{p}})$ after robust calibration

3.8 Tuning of hyperparameters

The cost to be optimized may involve, besides $\mathbf{p}$, some tuning parameters $\mathbf{q}$, called *hyperparameters*. Such is the case, for example, with *ridge estimators* (Remarks 4.10 and 5.3). Similarly, when the estimation problem is *ill posed*, *i.e.* when the estimate is too sensitive to small modifications of the data, one may use a regularized cost (see, *e.g.*, (Demoment, 1989; Thompson *et al.*, 1991))

$$j_r(\mathbf{p}, \mathbf{q}) = j_1(\mathbf{p}) + q_1 j_2(\mathbf{p}),$$

where j_1 is the cost associated with the initial ill-posed problem and the regularization function j_2 penalizes erratic variations of the signals computed by the model, for instance through the sum of the squares of their first or higher differences.

Such hyperparameters may be estimated by *cross-validation* (Golub, Heath and Wahba, 1979). Eliminating one datum $y(i)$ of $\mathbf{y}^s$, one can predict its value using the parameters $\hat{\mathbf{p}}_{-i}(\mathbf{q})$ estimated from the remaining data by minimizing $j_r(\mathbf{p}, \mathbf{q})$ with respect to **p**. The hyperparameters are then tuned to the value of **q** that provides the best average prediction over all possible eliminations, *i.e.* that minimizes

$$j_{\mathrm{cv}}(\mathbf{q}) = \frac{1}{n_{\mathrm{t}}}\sum_{i=1}^{n_{\mathrm{t}}} \{y(i) - y_{\mathrm{m}}[i, \hat{\mathbf{p}}_{-i}(\mathbf{q})]\}^2.$$

3.9 Conclusions

Any criterion is legitimate, insofar as it contributes to the fulfilment of the aims of the modelling, and many different criteria may be considered (see, *e.g.*, (Keating, Mason and Sen, 1993) for an interesting discussion on performance comparison for estimators). However, one should always state the options that have been taken, and justify them as far as possible. The use of sophisticated criteria based on specific hypotheses should be restricted to situations where these hypotheses can be relied upon or checked *a posteriori*. Otherwise, robust approaches should be preferred. Once the cost function j has been defined, it must be optimized. Note that optimization may be facilitated by the successive use of several cost functions. Sometimes, for example, the cost function derived by the maximum-likelihood approach has many local optimizers, at which local optimization techniques such as those presented in Section 4.3.3 may get trapped. A transient use of a quadratic cost may then lead to an estimate that can serve as a good starting point for a local optimization of a more sophisticated cost.

4 Optimization

The performance of a model structure, or of parameter estimates for a given model structure, is usually rated *via* a cost function j (Chapter 3). Finding the best possible model then corresponds to optimizing this cost. The resulting optimization problems *often* have the following characteristics:

— the number of parameters to be optimized is small, typically less than ten;
— the cost function is smooth, its first and second derivatives are relatively easy to compute;.
— optimization is unconstrained (although $\mathbf{p}$ might belong to some simple-shaped prior feasible set such as a box);
— the effects of the various parameters on the value of the cost are very unequal, *i.e.* the problem is ill conditioned;
— the problem is not convex and local optimizers may exist that do not correspond to the best possible value of the cost; we shall, however, see in Section 4.3.9.1 that suitable experimental conditions can sometimes eliminate such parasitic local optimizers.

The optimization may be *centred on the argument*. The cost function is then merely an intermediate in the search for $\hat{\mathbf{p}}$. This will be the case, for example, when estimating the parameters of a phenomenological model. All values of these parameters that lead to an acceptable value of the cost should then be searched for. Any possible singularity of the cost function in the neighbourhood of the optimum, or the possible existence of several global optimizers, should be taken into account.

The optimization may, on the other hand, be *centred on the cost*. All feasible global optimizers will then be equally acceptable. Such will often be the case, for instance, when estimating the parameters of a behavioural model.

Section 4.1 deals with cost functions that are quadratic with respect to an error affine in $\mathbf{p}$. Such affine errors occur for instance when an output error is used with an LP model structure (*i.e.* a structure such that the model output is linear in $\mathbf{p}$). It is then possible to derive explicit formulas for computing the global optimizer $\hat{\mathbf{p}}$, which correspond to the celebrated least-squares method.

The case where the cost is still quadratic in the error but the error is no longer affine in $\mathbf{p}$ is considered in Section 4.2. Various approaches involving an iterative application of the least-squares method are presented, and their limitations described.

Section 4.3 presents nonlinear programming techniques that can be used even when the cost function is not quadratic. In particular, constrained, non-differentiable, recursive and global optimization problems are considered.

Specific difficulties raised by the optimization of a process response are addressed in Section 4.4.

To simplify notation, the process output will most often be assumed to be scalar, but the methods described extend without difficulty to the vector case.

4.1 LP structures and quadratic cost functions

4.1.1 LP structures

The models considered can be written as

$$y_{\mathrm{m}}(t+1, \mathbf{p}) = \mathbf{r}^{\mathrm{T}}(t)\mathbf{p},$$

where $\mathbf{r}(t)$ is a vector of *known* quantities (therefore independent of $\mathbf{p}$), called the *regressor vector*. The independent variable t here takes integer values, and serves to index the various points where the model output must be computed. This does not imply that the measurement times are integer, or even that the independent variable corresponds to time. Static systems can therefore also be considered in this framework. Note that the fact of indexing y by $t + 1$ and $\mathbf{r}$ by t is in no way mandatory, but proves useful in the context of adaptive control.

The set of all outputs of the model $M(\mathbf{p})$, for t varying from one to n_{t}, can be written as

$$\mathbf{y}^{\mathrm{m}}(\mathbf{p}) = \begin{bmatrix} y_{\mathrm{m}}(1, \mathbf{p}) \\ \cdots \\ y_{\mathrm{m}}(n_{\mathrm{t}}, \mathbf{p}) \end{bmatrix} = \begin{bmatrix} \mathbf{r}^{\mathrm{T}}(0) \\ \cdots \\ \mathbf{r}^{\mathrm{T}}(n_{\mathrm{t}}-1) \end{bmatrix} \mathbf{p} = \mathbf{R}\mathbf{p}.$$

EXAMPLE 4.1

The parameters $\mathbf{p} = (a_1, \ldots, a_{n_{\mathrm{a}}}, b_0, \ldots, b_{n_{\mathrm{b}}})^{\mathrm{T}}$ of the model

$$y_{\mathrm{m}}(t, \mathbf{p}) = -\sum_{i=1}^{n_{\mathrm{a}}} a_i \frac{\mathrm{d}^i}{\mathrm{d}t^i} y(t) + \sum_{i=0}^{n_{\mathrm{b}}} b_i \frac{\mathrm{d}^i}{\mathrm{d}t^i} u(t),$$

with $n_{\mathrm{b}} \leq n_{\mathrm{a}}$, are to be estimated from input-output data recorded between $t = 0$ and $t = t_{\mathrm{f}}$, with a much smaller sampling period than the shortest time constant of the process, so they can be considered as recorded continuously. The model structure is LP, but trying to estimate its parameters through the computation of a regressor vector consisting of successive derivatives with respect to time of $y(t)$ and $u(t)$ is not to be recommended, because high-frequency noise will be dramatically amplified by repeated differentiation. The *modulating functions* approach can be used instead. The idea is to replace y and y_{m} by their scalar products with suitably chosen test functions ϕ_k

$$<y_{\mathrm{m}}|\phi_k> = -\sum_{i=1}^{n_{\mathrm{a}}} a_i <y^{(i)}|\phi_k> + \sum_{i=0}^{n_{\mathrm{b}}} b_i <u^{(i)}|\phi_k>,$$

with

$$f^{(i)} = \frac{\mathrm{d}^i}{\mathrm{d}t^i} f(t) \qquad \text{and} \qquad <f_1|f_2> = \int_0^{t_f} f_1(\tau) f_2(\tau)\,\mathrm{d}\tau.$$

Provided that ϕ_k is chosen sufficiently differentiable, with $\phi_k^{(i)}(0) = \phi_k^{(i)}(t_f) = 0$, $i = 0, \ldots, n_a - 1$, integration by parts can be used to carry differentiation over to the (analytically known) test functions

$$<y_m|\phi_k> = \sum_{i=1}^{n_a} (-1)^{i+1} a_i <y|\phi_k^{(i)}> + \sum_{i=0}^{n_b} (-1)^i b_i <u|\phi_k^{(i)}>.$$

The quantity $<y_m|\phi_k>$ is LP. One should use at least as many linearly independent test functions as there are parameters. The choice of these test functions is not trivial and obviously impacts on the estimate of **p** obtained (Richalet, Rault and Pouliquen, 1971). Provided that the model is LP, the method extends to equations involving pure delays, to multi-input-multi-output, non-LI or time-varying systems, and to partial differential equations (Loeb, 1967). It remains sensitive to low-frequency perturbations such as offsets, the influence of which is increased by integration. A partial solution is to introduce offset as an additional parameter. ◊

EXAMPLE 4.2

If the output of an LI model is computed by convolution of its impulse response h with its input u, discretized with period T:

$$y_m(t+1, \mathbf{p}) = T \sum_{i=1}^{n_p} h(i) u(t+1-i),$$

the parameters p_i might be the successive values of the discrete-time impulse response $h(i)$ $(i = 1, \ldots, n_p)$. The regressor vector then satisfies

$$\mathbf{r}(t) = [Tu(t), Tu(t-1), \ldots, Tu(t+1-n_p)]^{\mathrm{T}}. \qquad ◊$$

REMARKS 4.1

- The system must be such that $h(i)$ can be neglected for any $i > n_p$ (*Finite Impulse Response*, or FIR, models).
- Knowledge of the impulse response of an LI model makes it possible to simulate or control it just as well as any other representation. This approach, however, leads most often to overparametrization (about thirty parameters would typically be needed to describe the scalar impulse response of a third-order transfer function, which could be expressed with at most six parameters). As a result, a slight modification of the data may lead to a large change in the parameter estimates. This is the price paid for an LP model structure. Various *regularization* techniques can be implemented to give an acceptable solution to such *ill-posed* problems. They can be viewed as the introduction of additional prior information aimed at removing the ambiguity resulting from overparametrization. One may thus relate them to maximum *a posteriori* estimation; see Section 3.5.1 and (Demoment, 1989). ◊

EXAMPLE 4.2 (continued)

To obtain a smooth impulse response $h(t)$, one may characterize it as a linear combination of Laguerre functions (Wahlberg, 1991)

$$h(t, \mathbf{p}, \alpha) = \sum_{i=0}^{n_p-1} p_i L_i(t, \alpha),$$

where

$$L_i(t, \alpha) = \sqrt{2\alpha} \exp(-\alpha t) \sum_{k=0}^{i} \frac{i! \, (-2\alpha t)^k}{(i-k)! \, (k!)^2} .$$

These functions are orthonormal on [0, ∞], *i.e.*

$$\int_0^{\infty} L_i(\tau, \alpha) L_j(\tau, \alpha) \mathrm{d}\tau = \delta_{ij}.$$

For any given α, the output y_m becomes linear in $\mathbf{p}$. The resulting model may satisfactorily reproduce the behaviour of stable and non-oscillatory systems. Kautz functions, which include Laguerre functions as a special case, can be used to deal with oscillatory systems (Wahlberg, 1994). Generalizations of Laguerre and Kautz functions are considered in (Heuberger, Van den Hof and Bosgra, 1995).

The method extends to model impulse responses depending on some measurable external signal x, with the dependency on x incorporated in $\mathbf{p}$ (Velev, 1988),

$$h(t, \mathbf{p}(x), \alpha) = \sum_{i=0}^{n_p-1} p_i(x) L_i(t, \alpha).$$

Expanding each component of $\mathbf{p}$ in series, one defines a new set of parameters p_{ij} such that

$$p_i(x) = p_{i0} + p_{i1}x + p_{i2}x^2 + \ldots$$

and the model output remains linear in these new parameters. The model obtained remains stable for any value of x. ◊

EXAMPLE 4.3

Assume that the model satisfies the recurrence equation

$$y_m(t+1, \mathbf{p}) = -a_1 y_m(t, \mathbf{p}) - a_2 y_m(t-1, \mathbf{p}) - \ldots - a_{n_a} y_m(t+1-n_a, \mathbf{p})$$
$$+ b_1 u(t) + \ldots + b_{n_b} u(t+1-n_b),$$

where the parameters are the a_i's ($i = 1, \ldots, n_a$) and b_i's ($i = 1, \ldots, n_b$). This model structure is non-LP, since the right-hand side involves the product of parameters by model outputs that depend on **p**. If, on the other hand, the past values of the model output are replaced by the corresponding values of the output of the system studied, *another* structure is obtained (ARX if the noise is additive in the equation and corresponds to a sequence of i.i.d. random variables):

$$\begin{aligned} y_{\mathrm{mlp}}(t+1, \mathbf{p}) = &-a_1 y(t) - a_2 y(t-1) - \ldots - a_{n_\mathrm{a}} y(t+1-n_\mathrm{a}) \\ &+ b_1 u(t) + \ldots + b_{n_\mathrm{b}} u(t+1-n_\mathrm{b}). \end{aligned}$$

This structure is LP, provided that the inputs and outputs of the system do not depend on **p**. Its output $y_{\mathrm{mlp}}(t+1, \mathbf{p})$ can then be written as the scalar product of

$$\mathbf{p} = (a_1, a_2, \ldots, a_{n_\mathrm{a}}, b_1, b_2, \ldots, b_{n_\mathrm{b}})^\mathrm{T}$$

with

$$\mathbf{r}(t) = [-y(t), -y(t-1), \ldots, -y(t+1-n_\mathrm{a}), u(t), u(t-1), \ldots, u(t+1-n_\mathrm{b})]^\mathrm{T}. \qquad \Diamond$$

EXAMPLE 4.4

If the system considered can be described by a static nonlinearity followed by an LI dynamic part (Figure 4.1), a *Hammerstein* model structure may be employed.

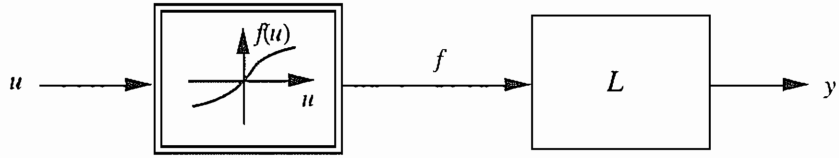


Figure 4.1. Static nonlinearity followed by a linear dynamic part

Approximating the nonlinearity by a polynomial in the scalar input u, one can write

$$A(q, \mathbf{p}) y_\mathrm{m}(t) = \sum_{i=1}^{n_\mathrm{B}} B_i(q, \mathbf{p}) u^i(t)$$

where $A(q, \mathbf{p})$ and $B_i(q, \mathbf{p})$ are polynomials in the unit delay operator q^{-1}

$$A(q, \mathbf{p}) = 1 + a_1 q^{-1} + \ldots + a_{n_\mathrm{a}} q^{-n_\mathrm{a}},$$

$$B_i(q, \mathbf{p}) = b_{1,i} q^{-1} + \ldots + b_{n_\mathrm{b},i} q^{-n_\mathrm{b}},$$

and where

$$\mathbf{p} = (a_1, \ldots, a_{n_\mathrm{a}}, b_{1,1}, \ldots, b_{n_\mathrm{b},1}, \ldots, b_{1,n_\mathrm{B}}, \ldots, b_{n_\mathrm{b},n_\mathrm{B}})^\mathrm{T}.$$

Substituting, as in Example 4.3, past values of the measured output y for the corresponding values of y_m, one obtains an LP (although non-LI) model structure, with a regressor vector given by

$$\mathbf{r}(t) = [-y(t), \ldots, -y(t+1-\mathrm{n_a}), u(t), \ldots, u(t+1-n_\mathrm{b}), u^2(t), \ldots, u^2(t+1-n_\mathrm{b}), \ldots, u^{n_\mathrm{B}}(t), \ldots, u^{n_\mathrm{B}}(t+1-n_\mathrm{b})]^\mathrm{T}. \qquad \Diamond$$

EXAMPLE 4.5

Another LP non-LI model structure corresponds to *bilinear* models, such as

$$y_\mathrm{m}(t, \mathbf{p}) = \sum_{i=1}^{n} p_i u(t-i) + \sum_{i=1}^{n} p_{n+i} y(t-i) + \sum_{i=1}^{n} \sum_{k=1}^{n} p_{i,k} u(t-i) y(t-k).$$

More generally, any model the output of which can be written as a polynomial in the past values of the inputs and measured process outputs could also be considered. ◊

4.1.2 Quadratic cost functions

The cost function to be minimized is assumed to satisfy

$$j(\mathbf{p}) = \mathbf{e}^\mathrm{T}(\mathbf{p})\mathbf{Q}\mathbf{e}(\mathbf{p}),$$

where **e** is some error, *e.g.* output error

$$\mathbf{e}(\mathbf{p}) = \mathbf{y}^\mathrm{s} - \mathbf{y}^\mathrm{m}(\mathbf{p}).$$

The weighting matrix **Q** is symmetrical and non-negative definite. It may have been chosen from hypotheses or knowledge about the measurement noise (Chapter 3). If, for instance, the data are assumed to have been generated by

$$\mathbf{y}^\mathrm{s} = \mathbf{y}^\mathrm{m}(\mathbf{p}^*) + \mathbf{n},$$

where **n** is a realization of a Gaussian random vector with zero mean and known covariance $\mathbf{\Sigma}$, the maximum-likelihood method suggests $\mathbf{Q} = \mathbf{\Sigma}^{-1}$. **Q** may also be chosen in a more heuristic manner. In the absence of any specific information, **Q** is often taken as the identity matrix.

4.1.3 Least-squares estimator

The *least-squares estimator* $\hat{\mathbf{p}}_\mathrm{ls}$, developed by Gauss and Legendre at the beginning of the 19th century, minimises the quadratic cost

$$j(\mathbf{p}) = \mathbf{e}^\mathrm{T}(\mathbf{p})\mathbf{Q}\mathbf{e}(\mathbf{p}),$$

under the constraint that the error is affine in the parameters, which will be true for an output error with an LP model structure

$$\mathbf{e}(\mathbf{p}) = \mathbf{y}^\mathrm{s} - \mathbf{R}\mathbf{p}.$$

The first derivative of the cost with respect to $\mathbf{p}$ is zero at the optimum. Since $\mathbf{Q}$ is symmetrical, this implies that

$$\frac{\partial j}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\hat{\mathbf{p}}_{\mathrm{ls}}} = -2\mathbf{R}^{\mathrm{T}}\mathbf{Q}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}\hat{\mathbf{p}}_{\mathrm{ls}}) = \mathbf{0}.$$

Provided that $\mathbf{R}^{\mathrm{T}}\mathbf{Q}\mathbf{R}$ is invertible, the least-squares estimate is therefore given by

$$\hat{\mathbf{p}}_{\mathrm{ls}} = (\mathbf{R}^{\mathrm{T}}\mathbf{Q}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{Q}\mathbf{y}^{\mathrm{s}}.$$

4.1.3.1 Properties of the least-squares estimator

PLS1: The vector $\hat{\mathbf{p}}_{\mathrm{ls}}$ of the parameter estimates is obtained *analytically* from the data $\mathbf{y}^{\mathrm{s}}$, matrix of regressors $\mathbf{R}$ and weighting matrix $\mathbf{Q}$.

PLS2: It is trivial to check that $\mathbf{e}^{\mathrm{T}}(\hat{\mathbf{p}}_{\mathrm{ls}})\mathbf{Q}\mathbf{y}^{\mathrm{m}}(\hat{\mathbf{p}}_{\mathrm{ls}}) = 0$. If $\mathbf{Q} = \mathbf{I}$, this implies that the error at the optimum is orthogonal to the model output; $\mathbf{y}^{\mathrm{m}}(\hat{\mathbf{p}}_{\mathrm{ls}})$ is thus the orthogonal projection of $\mathbf{y}^{\mathrm{s}}$ onto the locus of all possible model responses, *i.e.* onto the *expectation surface* $\mathbb{S}_{\mathrm{exp}} = \{\mathbf{y}^{\mathrm{m}}(\mathbf{p}), \mathbf{p} \in \mathbb{R}^{n_{\mathrm{p}}}\}$ (Figure 4.2). Since the model structure considered here is LP, the expectation surface is a hyperplane, but this is not always so, as will be seen in Chapters 5 and 6. The orthogonal projection operator is given here by $\Pi = \mathbf{R}(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}$.

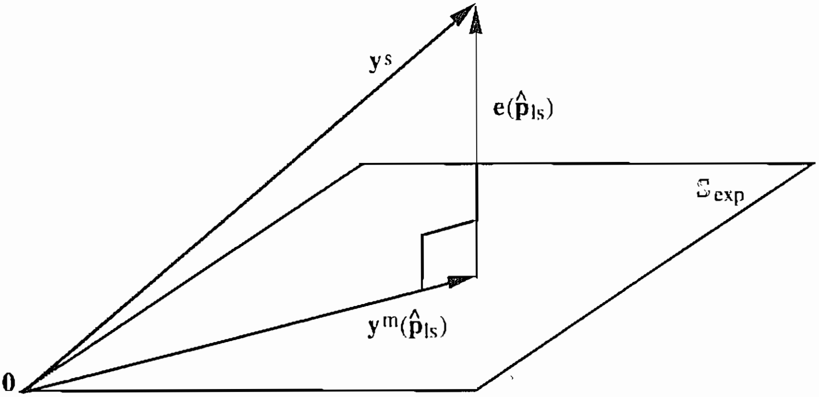

$\mathbf{y}^{\mathrm{s}}$
$\mathbf{e}(\hat{\mathbf{p}}_{\mathrm{ls}})$
$\mathbb{S}_{\mathrm{exp}}$
$\mathbf{y}^{\mathrm{m}}(\hat{\mathbf{p}}_{\mathrm{ls}})$
0

Figure 4.2. At the optimum, the error and model output are orthogonal

PLS3: The matrix $\mathbf{R}^{\mathrm{T}}\mathbf{Q}\mathbf{R}$ to be inverted is symmetrical and $n_{\mathrm{p}} \times n_{\mathrm{p}}$ (with $n_{\mathrm{p}} = \dim \mathbf{p}$). Its dimensions do not, therefore, depend on the number of data points. Except for pathological cases where the information content of the inputs is too poor or the model is unidentifiable, it is positive-definite and therefore invertible. If not, one might use a model with fewer parameters.

PLS4: Even if $\hat{\mathbf{p}}_{\mathrm{ls}}$ turns out not to be the best possible estimate of $\mathbf{p}$ given the aim of the modelling, it may serve to *initialize* some iterative search with another cost function and/or another model better suited to the original problem.

PLS5: The matrix $(\mathbf{R}^{\mathrm{T}}\mathbf{Q}\mathbf{R})^{-1}$ contains important information on the performance of the estimator (Section 5.3). It is therefore useful to examine it, or at least its diagonal entries. The smaller these entries are, the better the precision of the estimated parameters is, which suggests criteria for choice of the regressor vectors (Chapter 6).

EXAMPLE 4.6

The following data have been obtained by sampling a process output at $t = 1, 2, 3$:

$$y(1) = 270,\ y(2) = 36,\ y(3) = 5.$$

The process output is to be described by

$$y_{\mathrm{m}}(t, \mathbf{p}) = p_1 \exp(-p_2 t).$$

As it stands, this model structure is not LP, so the least-squares method does not apply. An LP model structure can however be obtained by performing a logarithmic transformation on the data and model output:

$$y'(t) = \ln y(t),$$

$$y_{\mathrm{m}}'(t, \mathbf{q}) = \ln y_{\mathrm{m}}(t, \mathbf{p}) = \ln p_1 - p_2 t,$$

and taking the parameters to be estimated as

$$q_1 = \ln p_1 \quad \text{and} \quad q_2 = p_2.$$

The vectors of all transformed process and model outputs can be written as

$$\mathbf{y}^{\mathrm{s}'} = \begin{bmatrix} \ln 270 \\ \ln 36 \\ \ln 5 \end{bmatrix} \approx \begin{bmatrix} 5.6 \\ 3.6 \\ 1.6 \end{bmatrix} \quad \text{and} \quad \mathbf{y}^{\mathrm{m}'}(\mathbf{q}) = \begin{bmatrix} 1 & -1 \\ 1 & -2 \\ 1 & -3 \end{bmatrix} \mathbf{q} = \mathbf{R}\mathbf{q}.$$

The least-squares estimate of $\mathbf{q}$ is therefore

$$\hat{\mathbf{q}}_{\mathrm{ls}} = (\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{y}^{\mathrm{s}'} \approx \begin{bmatrix} 7.6 \\ 2 \end{bmatrix},$$

which corresponds to

$$\hat{p}_1 = \exp \hat{q}_1 \approx 1970 \quad \text{and} \quad \hat{q}_2 = \hat{p}_2 \approx 2.$$

Running the corresponding model, one gets

$$\mathbf{y}^{\mathrm{m}} \approx \begin{bmatrix} 268 \\ 36.5 \\ 5 \end{bmatrix}, \text{ compared with } \mathbf{y}^{\mathrm{s}} = \begin{bmatrix} 270 \\ 36 \\ 5 \end{bmatrix}.$$

Note that $\hat{\mathbf{p}}$ is not the best estimate of $\mathbf{p}$ in the least-squares sense, since the definition of the error has been changed to make it affine in the parameters. One may hope, however, that $\hat{\mathbf{p}}$ is close enough to this best estimate to provide a good starting point for an iterative search. More general transformations are considered, *e.g.*, in (Box and Cox, 1964; Atkinson, 1985, 1995). ◊

4.1.3.2 Numerical considerations

Except when treating small academic problems, one should avoid using the above explicit formula for $\hat{\mathbf{p}}_{\text{ls}}$, which turns out to be numerically disastrous. For ill-conditioned problems, *i.e.* when $\mathbf{R}^{\text{T}}\mathbf{Q}\mathbf{R}$ is almost singular, much more accurate results are obtained by *singular-value decomposition*; see, *e.g.*, (Klema and Laub, 1980). In this approach, $\mathbf{R}$ is factorized as $\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^{\text{T}}$, where $\mathbf{W}$ is a diagonal $n_{\text{p}} \times n_{\text{p}}$ matrix and $\mathbf{U}^{\text{T}}\mathbf{U} = \mathbf{V}^{\text{T}}\mathbf{V} = \mathbf{I}_{n_{\text{p}}}$. The diagonal entries of $\mathbf{W}$ are the *singular values* $w_{i,i}$ of $\mathbf{R}$ ($w_{i,i} \geq 0$). All decent scientific software libraries include a subroutine performing this factorization, and one should refrain from attempting to code it again! $\mathbf{R}^{\text{T}}\mathbf{Q}\mathbf{R}$ will be singular if and only if $\mathbf{R}$ has at least one zero singular value. Most often, however, no singular value is strictly zero, and the *condition number* of $\mathbf{R}$ is defined as the ratio ρ of its largest singular value to its smallest. $\mathbf{R}$ will be said to be ill-conditioned as soon as $1/\rho$ is smaller than the accuracy of the floating-point representation used (about 10^{-16} in double precision according to the ANSI/IEEE 754-1985 norm, which most present-day computers comply with). If $\mathbf{Q} = \mathbf{I}$, the matrix to be inverted to compute $\hat{\mathbf{p}}_{\text{ls}}$ can be written as

$$\mathbf{R}^{\text{T}}\mathbf{R} = \mathbf{V}\mathbf{W}\mathbf{U}^{\text{T}}\mathbf{U}\mathbf{W}\mathbf{V}^{\text{T}} = \mathbf{V}\mathbf{W}^2\mathbf{V}^{\text{T}}.$$

This implies that the condition number of $\mathbf{R}^{\text{T}}\mathbf{R}$ is equal to the square of that of $\mathbf{R}$, and explains the danger involved in the computation of $\hat{\mathbf{p}}_{\text{ls}}$ by the analytical formula.

Once singular-value decomposition of $\mathbf{R}$ has been performed, it must be used to compute $\hat{\mathbf{p}}_{\text{ls}}$. Assume first that $\mathbf{Q} = \mathbf{I}$. If $\mathbf{R}$ is square and invertible,

$$\hat{\mathbf{p}}_{\text{ls}} = \mathbf{R}^{-1}\mathbf{y}^{\text{s}} = \mathbf{V}\ [\text{diag}(\frac{1}{w_{i,i}})]\ \mathbf{U}^{\text{T}}\mathbf{y}^{\text{s}}.$$

If the dimension of $\mathbf{y}^{\text{s}}$ is larger than that of $\mathbf{p}$ and no singular value is zero, $\hat{\mathbf{p}}_{\text{ls}}$ is unique, and also given by

$$\hat{\mathbf{p}}_{\text{ls}} = \mathbf{V}\ [\text{diag}(\frac{1}{w_{i,i}})]\ \mathbf{U}^{\text{T}}\mathbf{y}^{\text{s}}.$$

If some columns of $\mathbf{R}$ are linearly dependent, then at least one of its singular values $w_{i,i}$ is zero and there exist an infinity of estimates $\hat{\mathbf{p}}_{\text{ls}}$ that are equivalent in the sense of the least-squares criterion. Among these, the smallest in Euclidean norm is again obtained with the formula above, provided that all terms $1/w_{i,i}$ associated with $w_{i,i} = 0$ are replaced by 0. Similarly, if some $w_{i,i}$'s are very small, the estimator tends to drag $\hat{\mathbf{p}}_{\text{ls}}$ very far along directions that "almost" belong to the kernel of $\mathbf{R}$. Surprising as it may seem, one then gets better values of the cost by replacing the terms $1/w_{i,i}$ corresponding to these very small $w_{i,i}$'s by zero (Press *et al.*, 1986). A threshold remains to be set to decide which singular values will be considered very small.

If $\mathbf{Q} \neq \mathbf{I}$, the same equation can be used for the computation of $\hat{\mathbf{p}}_{\text{ls}}$, provided that $\mathbf{Q}$ is factorized as $\mathbf{Q} = \mathbf{M}^{\text{T}}\mathbf{M}$ and $\mathbf{e}(\mathbf{p})$ is replaced by $\mathbf{M}\mathbf{e}(\mathbf{p})$, which amounts to replacing $\mathbf{y}^{\text{s}}$ by $\mathbf{M}\mathbf{y}^{\text{s}}$ and $\mathbf{R}$ by $\mathbf{M}\mathbf{R}$.

REMARK 4.2

There is a vast literature about the least-squares method and the connected problem of computing a *pseudo-inverse* of the matrix $\mathbf{R}$ (Albert, 1972). Equality or inequality constraints in particular can be taken into account. For more details, see (Lawson and Hanson, 1974) and Section 4.3.4.2. ◊

4.1.4 Data-recursive least squares

Whereas the algorithm presented in the previous section processes all data collected on the system as a batch (*off-line*), its recursive counterpart processes them one after the other (*on-line*). This approach may be preferred for either of two reasons, which lead to different policies.

— There may be too many data for them to be stored simultaneously in the computer. One then wishes to memorize a limited amount of information, independent of the number of data points, rather than having to manage a large data base. In this case, $\mathbf{p}^*$ is considered fixed, and one wishes to obtain the same result as with the non-recursive algorithm.

— One may wish to use the results of the identification to take immediate decisions from the measurements performed so far, without having to wait until all data have been collected. One may, for instance, wish to track the parameters of a system to make sure that they remain in their normal operating range (*fault detection and diagnosis*), or compute a control law based on the most recent estimates of the parameters available (*adaptive control*). It is then assumed that $\mathbf{p}^*$ may vary with t.

4.1.4.1 p* assumed to be constant

Let $\hat{\mathbf{p}}_{\rm ls}(t)$ be the estimate of $\mathbf{p}^*$ obtained by non-recursive least squares from all data available up to time t. If we assume that $\mathbf{Q} = \mathbf{I}$ to simplify notation, we can write

$$\hat{\mathbf{p}}_{\rm ls}(t) = [\mathbf{R}^{\rm T}(t-1)\mathbf{R}(t-1)]^{-1}\mathbf{R}^{\rm T}(t-1)\mathbf{y}^{\rm s}(t),$$

where

$$\mathbf{R}(t-1) = \begin{bmatrix} \mathbf{r}^{\rm T}(0) \\ \mathbf{r}^{\rm T}(1) \\ \cdots \\ \mathbf{r}^{\rm T}(t-1) \end{bmatrix} \quad \text{and} \quad \mathbf{y}^{\rm s}(t) = \begin{bmatrix} y(1) \\ y(2) \\ \cdots \\ y(t) \end{bmatrix}.$$

At time $t+1$, a new measurement $y(t+1)$ is collected, so

$$\mathbf{R}(t) = \begin{bmatrix} \mathbf{R}(t-1) \\ \mathbf{r}^{\rm T}(t) \end{bmatrix} \quad \text{and} \quad \mathbf{y}^{\rm s}(t+1) = \begin{bmatrix} \mathbf{y}^{\rm s}(t) \\ y(t+1) \end{bmatrix}.$$

We wish to update $\hat{\mathbf{p}}_{\mathrm{ls}}$ to take this new information into account, but without having to store $\mathbf{R}$ and $\mathbf{y}^{\mathrm{s}}$ of ever-increasing size in the computer. Let us write $\hat{\mathbf{p}}_{\mathrm{ls}}$ as a function of the sequence of regressor vectors $\mathbf{r}$ and output measurements y:

$$\hat{\mathbf{p}}_{\mathrm{ls}}(t) = \left[\sum_{i=1}^{t} \mathbf{r}(i-1)\mathbf{r}^{\mathrm{T}}(i-1)\right]^{-1}\left[\sum_{i=1}^{t} \mathbf{r}(i-1)y(i)\right].$$

Define the matrix $\mathbf{M}(t)$ and vector $\mathbf{v}(t)$ by

$$\mathbf{M}(t) = \sum_{i=1}^{t} \mathbf{r}(i-1)\mathbf{r}^{\mathrm{T}}(i-1) \quad \text{and} \quad \mathbf{v}(t) = \sum_{i=1}^{t} \mathbf{r}(i-1)y(i).$$

Then

$$\hat{\mathbf{p}}_{\mathrm{ls}}(t) = \mathbf{M}^{-1}(t)\mathbf{v}(t),$$

which can be put into a recursive form without any approximation, since

$$\mathbf{M}(t+1) = \mathbf{M}(t) + \mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t),$$

$$\mathbf{v}(t+1) = \mathbf{v}(t) + \mathbf{r}(t)y(t+1)$$

and

$$\hat{\mathbf{p}}_{\mathrm{ls}}(t+1) = \mathbf{M}^{-1}(t+1)\mathbf{v}(t+1).$$

It therefore suffices to store the present values of $\mathbf{M}$ and $\mathbf{v}$, which obviously requires much less memory than storing $\mathbf{R}$ and $\mathbf{y}^{\mathrm{s}}$. Now express $\hat{\mathbf{p}}_{\mathrm{ls}}(t+1)$ as a function of $\hat{\mathbf{p}}_{\mathrm{ls}}(t)$:

$$\hat{\mathbf{p}}_{\mathrm{ls}}(t+1) = \mathbf{M}^{-1}(t+1)[\mathbf{v}(t) + \mathbf{r}(t)y(t+1)] = \mathbf{M}^{-1}(t+1)[\mathbf{M}(t)\hat{\mathbf{p}}_{\mathrm{ls}}(t) + \mathbf{r}(t)y(t+1)].$$

Replace $\mathbf{M}(t)$ by its expression as a function of $\mathbf{M}(t+1)$,

$$\hat{\mathbf{p}}_{\mathrm{ls}}(t+1) = \mathbf{M}^{-1}(t+1)\{[\mathbf{M}(t+1) - \mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t)]\hat{\mathbf{p}}_{\mathrm{ls}}(t) + \mathbf{r}(t)y(t+1)\}.$$

After expansion and simplification, we get

$$\hat{\mathbf{p}}_{\mathrm{ls}}(t+1) = \hat{\mathbf{p}}_{\mathrm{ls}}(t) + \mathbf{M}^{-1}(t+1)\mathbf{r}(t)[y(t+1) - \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}_{\mathrm{ls}}(t)],$$

where one can recognize:

— a one-step-ahead prediction of $y(t+1)$ by $y_{\mathrm{m}}(t+1) = \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}_{\mathrm{ls}}(t)$,
— the prediction error $e_{\mathrm{p}}(t+1, \hat{\mathbf{p}}_{\mathrm{ls}}(t)) = y(t+1) - \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}_{\mathrm{ls}}(t)$,
— a vector correction gain $\mathbf{k}(t+1) = \mathbf{M}^{-1}(t+1)\mathbf{r}(t)$, often called the Kalman gain for reasons to become apparent in Section 4.1.6.

Therefore

$$\hat{\mathbf{p}}_{\mathrm{ls}}(t+1) = \hat{\mathbf{p}}_{\mathrm{ls}}(t) + (\text{vector correction gain}) \times (\text{prediction error}),$$

which implies that the parameters do not change when the prediction error is zero. As it now stands, the algorithm still requires the inversion of $\mathbf{M}(t+1)$ to take $y(t+1)$ into account. This can be avoided with the help of the *matrix inversion lemma*

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1},$$

where all matrices inverted are assumed to be invertible. Define $\mathbf{P}(t+1)$ as

$$\mathbf{P}(t+1) = \mathbf{M}^{-1}(t+1) = [\mathbf{M}(t) + \mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t)]^{-1},$$

and set $\mathbf{A} = \mathbf{M}(t) = \mathbf{P}^{-1}(t)$, $\mathbf{B} = \mathbf{r}(t)$, $\mathbf{C} = 1$ and $\mathbf{D} = \mathbf{r}^{\mathrm{T}}(t)$. The lemma then implies that

$$\mathbf{P}(t+1) = \mathbf{P}(t) - \mathbf{P}(t)\mathbf{r}(t)[1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)]^{-1}\mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t).$$

At each step, the inversion of a matrix has thus been replaced by a division by the *scalar* $[1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)]$. The correction gain can now be written as

$$\mathbf{k}(t+1) = \mathbf{M}^{-1}(t+1)\mathbf{r}(t) = \mathbf{P}(t+1)\mathbf{r}(t) = \mathbf{P}(t)\mathbf{r}(t)\left[1 - \frac{\mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)}{1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)}\right]$$

$$= \frac{\mathbf{P}(t)\mathbf{r}(t)}{1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)}.$$

Iteration. In summary, one iteration of recursive least squares consists of the following steps. Before $y(t+1)$ is collected:

— compute the correction gain

$$\mathbf{k}(t+1) = \frac{\mathbf{P}(t)\mathbf{r}(t)}{1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)};$$

— update $\mathbf{P}$, which can also be written as

$$\mathbf{P}(t+1) = \mathbf{P}(t) - \mathbf{k}(t+1)\mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t);$$

— predict the output

$$y_{\mathrm{m}}(t+1) = \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}_{\mathrm{ls}}(t).$$

After $y(t+1)$ has been collected:

— update the parameter estimates according to

$$\hat{\mathbf{p}}_{\mathrm{ls}}(t+1) = \hat{\mathbf{p}}_{\mathrm{ls}}(t) + \mathbf{k}(t+1)[y(t+1) - y_{\mathrm{m}}(t+1)].$$

This shortens the delay between the measurement of $y(t+1)$ and the updating of the estimates.

Initialization. In principle, one should wait until enough data have been collected for $\mathbf{M}(t)$ to become invertible, and initialize the algorithm by $\mathbf{P}(t) = \mathbf{M}^{-1}(t)$ and $\hat{\mathbf{p}}_{\mathrm{ls}}(t) = \mathbf{P}(t)\mathbf{v}(t)$. Recursive least-squares estimation is then *strictly equivalent* to its off-line counterpart, up to numerical errors introduced by finite-precision arithmetic. No information has been lost, although past data have been dropped.

In practice, however, one often chooses $\hat{\mathbf{p}}_{\mathrm{ls}}(0)$ arbitrarily, *e.g.* $\hat{\mathbf{p}}_{\mathrm{ls}}(0) = \mathbf{0}$, and $\mathbf{P}(0)$ as an identity matrix multiplied by some large positive number, *e.g.* $\mathbf{P}(0) = 10^{12}\,\mathbf{I}_{n_\mathrm{p}}$. This amounts to saying that no confidence is bestowed on this initial estimate for the parameters, because $\mathbf{P}$ corresponds to $[\mathbf{R}^\mathrm{T}\mathbf{R}]^{-1}$ of the off-line least-squares method, which characterizes the uncertainty in the estimated parameters (Chapter 5). $\mathbf{P}$ then decreases very quickly in the initial phase, so the values of $\hat{\mathbf{p}}_{\mathrm{ls}}(0)$ and $\mathbf{P}(0)$ are not critical.

Properties of the data-recursive least-squares estimator

PRLS1: The off-line method required the invertibility of $\mathbf{R}^\mathrm{T}\mathbf{Q}\mathbf{R}$, *i.e.* the identifiability of the model from the actual data collected. Since recursive least-squares estimation no longer involves matrix inversion, it will not be able to detect unidentifiability and will converge to a particular solution that depends on the initialization.

PRLS2: Since

$$\mathbf{P}(t+1) - \mathbf{P}(t) = -\frac{\mathbf{P}(t)\mathbf{r}(t)\mathbf{r}^\mathrm{T}(t)\mathbf{P}(t)}{1 + \mathbf{r}^\mathrm{T}(t)\mathbf{P}(t)\mathbf{r}(t)}\,,$$

the difference between two successive values of $\mathbf{P}$ is negative semi-definite. In this sense, $\mathbf{P}$ can therefore only decrease (or stay constant if $\mathbf{r}(t) = \mathbf{0}$, which means that $y(t+1)$ brings no information on $\mathbf{p}$). The information available about the system studied increases with time, so the uncertainty in the parameters decreases. For a given value of the regressor vector, this implies that the correction gain $\mathbf{k}(t+1)$ decreases. The prediction errors are thus less and less taken into account to adjust the model, because they are more and more attributed to noise.

PRLS3: The previous equation implies that if $\mathbf{P}(t)$ becomes singular for numerical reasons, it will stay so forever. To overcome this difficulty, one may add $\varepsilon\mathbf{I}_{n_\mathrm{p}}$ to $\mathbf{P}(t)$, where ε is some small positive number. This idea, due to Levenberg and Marquardt, will be encountered again in Section 4.3.3.5. Another possible approach is to take advantage of the fact that $\mathbf{P}(t)$ should be symmetric and positive-definite to compute it in a factorized form. Bierman's U-D factorization algorithm uses a normalized Cholesky decomposition $\mathbf{P}(t) = \mathbf{U}(t)\mathbf{D}(t)\mathbf{U}^\mathrm{T}(t)$, where $\mathbf{U}(t)$ is an upper triangular matrix, with all diagonal entries equal to one, and $\mathbf{D}(t)$ is a diagonal matrix. The computation is only marginally more complicated than in the initial algorithm. For more details, see (Bierman, 1977; Ljung and Söderström, 1983).

PRLS4: Since $\mathbf{P}$ characterizes the uncertainty in the estimated parameters, it is interesting to monitor its evolution, or at least that of its diagonal terms. This may help in choosing the later regressor vectors (Section 6.3.2.2).

PRLS5: If the sequence of regressor vectors is known *a priori*, the associated sequences $\{\mathbf{P}(t)\}$ and $\{\mathbf{k}(t)\}$ can be computed off-line, before any measurement takes place. The actual sequence of prediction errors has therefore no influence on the correction gains, and one is strongly advised to monitor it in order to detect any possible divergence of the algorithm.

4.1.4.2 $\mathbf{p}^*$ may drift

If the parameters $\mathbf{p}^*$ may slowly vary, *e.g.* because an LI model is used to describe a non-LI process around some changing operating point, one wishes $\hat{\mathbf{p}}_{\mathrm{ls}}$ to track these variations "in real time". This requires *forgetting measurements that are too old*, because they correspond to an out-of-date situation and would distort estimation. A particularly simple technique for this purpose is *exponential forgetting*, which weights prediction errors in the cost function exponentially, decreasing with time elapsed:

— present time $t + 1$ receives unit weight,
— past time $t + 1 - n$ is weighted by λ^n,

with λ the *forgetting factor*, such that $0 < \lambda \leq 1$. If $\lambda = 1$, no forgetting takes place, and the smaller λ is, the more quickly the past is forgotten. To implement this policy, it suffices to update $\mathbf{M}$ and $\mathbf{v}$ according to

$$\mathbf{M}(t+1) = \lambda\mathbf{M}(t) + \mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t) \qquad \text{and} \qquad \mathbf{v}(t+1) = \lambda\mathbf{v}(t) + \mathbf{r}(t)y(t+1),$$

which amounts to multiplying the past values by λ before adding the present contribution. By the same procedure as when $\lambda = 1$, one gets

$$\mathbf{k}(t+1) = \frac{\mathbf{P}(t)\mathbf{r}(t)}{\lambda + \mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)},$$

$$\mathbf{P}(t+1) = \frac{1}{\lambda}\,[\mathbf{P}(t) - \mathbf{k}(t+1)\mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)],$$

$$y_{\mathrm{m}}(t+1) = \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}_{\mathrm{ls}}(t),$$

$$\hat{\mathbf{p}}_{\mathrm{ls}}(t+1) = \hat{\mathbf{p}}_{\mathrm{ls}}(t) + \mathbf{k}(t+1)[y(t+1) - y_{\mathrm{m}}(t+1)].$$

Computation is therefore no more complicated than without forgetting.

Properties of recursive least squares with exponential forgetting

PRLSEF1: It is trivial to check that when $\lambda = 1$ the equations become those obtained without forgetting.

PRLSEF2: As when no forgetting takes place, if $\mathbf{P}$ becomes singular, it will stay so. Forgetting will just make this event more likely. It is therefore necessary to modify the algorithm to prevent this. One may again either add $\varepsilon\mathbf{I}_{n_{\mathrm{p}}}$ to $\mathbf{P}$, with ε a small positive number, or use U-D factorization.

PRLSEF3: If $\lambda < 1$, $\mathbf{P}$ may increase with time, contrary to what happened without forgetting. If, in particular, $\mathbf{r}(t) = \mathbf{0}$ then $\mathbf{P}(t+1) = \mathbf{P}(t)/\lambda > \mathbf{P}(t)$. When this situation persists, it entails a definite risk of explosion. A possible way out is to use a variable forgetting factor $\lambda(t)$, such that the trace of $\mathbf{P}(t+1)$ remains below some given bound.

PRLSEF4: When $\lambda < 1$, $\mathbf{P}(t+1)$ and $\mathbf{k}(t+1)$ no longer tend to $\mathbf{0}$. The unavoidable prediction errors therefore always cause a modification of the estimates, so $\hat{\mathbf{p}}_{\mathrm{ls}}$ no longer converges to a constant value. As a result, small meaningless prediction errors, corresponding for instance to a system almost at equilibrium, may in practice cause a drift of the parameter estimates. It is therefore necessary to freeze the algorithm when the prediction error becomes small enough.

PRLSEF5: The smaller λ is, the larger the correction gain remains, which increases the tracking capability of the algorithm, at the cost of amplifying the random fluctuations of the parameter estimates. A compromise must therefore be drawn, most often empirically, which is relatively easy since λ is a scalar (typically, $\lambda > 0.95$).

4.1.4.3 $\mathbf{p}^*$ may jump

When the parameters to be estimated are known to vary by jumps, exponential forgetting becomes unsuitable, because it combines erratic moves of the estimated parameters (due to the fact that the correction gain does not tend to zero) and slow adaptation when jumps occur (due to the fact that λ must be taken close to one to ensure stability). Better results should then be obtained by using recursive least squares without forgetting, monitoring the prediction error and increasing $\mathbf{P}$ as soon as this error becomes too unlikely. This amounts to admitting that the uncertainty in the estimated parameters has increased.

To decide whether the prediction error on $y(t+1)$ is likely, one may use the fact that its variance is given by $\sigma^2[1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)]$ when the parameters are adapted, where σ^2 is the variance of the noise in the observations.

4.1.4.4 Application to adaptive control

Consider the scheme of Figure 4.3, which involves two models and two optimization algorithms. The *reference model* generates the reference trajectory $\mathbf{y}_{\mathrm{r}}$, expressing how the process should behave. It should not be confused with the *model of the process*, which generates the model output $\mathbf{y}_{\mathrm{m}}$, expressing how the process is believed to behave in reality. The first optimization algorithm estimates the parameters of the process, whereas the second one computes the control law to be applied. Assume that we known how to adapt the parameters of the process model so as to ensure that $\mathbf{y}_{\mathrm{m}}(t) \rightarrow \mathbf{y}(t)$. If the input $\mathbf{u}$ is chosen to impose $\mathbf{y}_{\mathrm{m}}(t) \rightarrow \mathbf{y}_{\mathrm{r}}(t)$, this will entail that $\mathbf{y}(t) \rightarrow \mathbf{y}_{\mathrm{r}}(t)$. A key point is to guarantee the stability of the resulting complex nonlinear feedback system.

REMARKS 4.3

— What is controlled is the output of the model of the process, and not that of the process itself. It is only because the model output is constrained to resemble that of the process that the process output will resemble the reference trajectory.

— Even if the structure of the process and its model are identical and y and y_{m} both converge to y_{r}, this does not imply that $\hat{\mathbf{p}}$ converges to $\mathbf{p}^*$, because the input signal may be too poor to make the parameters identifiable in practice.

— Provided that the variation of $\mathbf{p}^*$ is slow enough, the parameter estimates obtained, *e.g.* by recursive least squares with exponential forgetting, are close to the best estimate of $\mathbf{p}^*$ in the least-squares sense *for the input actually applied to the system*. If this input is rich enough for the output to contain much information on $\mathbf{p}^*$, this may be considered an advantage, because the model is always an approximation of reality and the best approximation depends on the type of input considered. The model obtained is thus suited to the operating conditions of the system. If, on the other hand, the information content of the process output is very small, for instance because the input is almost constant, then the parameters may become completely erroneous although the prediction error remains negligible. This may result in very bad transient behaviour following the next change of operating point.

— When intermittent perturbations act within the passband of the process, the algorithm will modify the model parameters to try to reduce the prediction error, although $\mathbf{p}^*$ has not changed. This may result in erratic behaviour of the control system. It therefore seems advisable to test the performance of adaptive control systems in realistic simulation conditions, including perturbations, before implementation is considered. This implementation may require the combining of ideas from adaptive and robust control theories; see, *e.g.*, (Irving, Daoudi and Bourlès, 1991). ◊

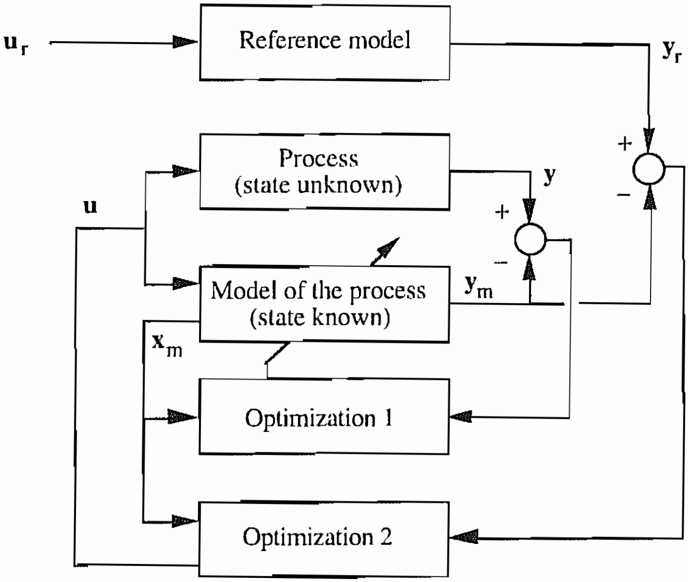


Figure 4.3. Example of model-reference adaptive control

Figure 4.3 illustrates a possible model-reference adaptive control scheme. It involves a parallel model of the process, but series or series-parallel models could be used as well. Moreover, many alternative criteria and algorithms could be employed to estimate the parameters and compute the control law. It is thus possible to create a very large number of adaptive control schemes (Chalam (1987) quotes more than 1200 references!). More information can be found, *e.g.*, in (Landau, 1979; Goodwin and Sin, 1984; Åström and Wittenmark, 1984; Kumar and Varaiya, 1986; Bitmead, Gevers and Wertz, 1990; Isermann, Lachmann and Matko, 1992; Kaufman, Bar-Kana and Sobel, 1994) and in the countless papers devoted to the subject in international journals and conferences. Here, we shall only mention one of the existing approaches, because of its direct links with recursive least squares. This approach was proposed by Åström and Wittenmark (1973) under the name of *Self TUning REgulator* (STURE), and has been implemented on many industrial processes (Åström *et al.*, 1977).

STURE. The process is described by an LP prediction model (here one step ahead)

$$y_m(t+1, \mathbf{p}) = -a_1 y(t) - a_2 y(t-1) - \dots - a_{n_a} y(t+1-n_a) + b_1 u(t) + \dots + b_{n_b} u(t+1-n_b).$$

Optimization algorithm 1 estimates the parameters $\mathbf{p}$ of the process model by recursive least squares with exponential forgetting. Optimization algorithm 2, which computes the control law, may implement various policies. One may, for instance, compute an optimal control sequence if this can be performed in real time. At each step, one will then apply the first entry of this sequence to the process (sliding-horizon optimal control). This is the basic idea of *generalized predictive control* (Clarke, Mohtadi and Tuffs, 1987; Clarke, 1988; Bitmead, Gevers and Wertz, 1990). A cruder policy is to compute the control $u(t)$ so as to make the predicted output $y_m(t+1, \hat{\mathbf{p}})$ equal to the corresponding value of the reference trajectory $y_r(t+1)$, *i.e.* to ensure

$$y_r(t+1) = -\hat{a}_1 y(t) - \hat{a}_2 y(t-1) - \dots - \hat{a}_{n_a} y(t+1-n_a) + \hat{b}_1 u(t) + \dots + \hat{b}_{n_b} u(t+1-n_b),$$

where $\hat{a}_i$ and $\hat{b}_i$ are entries of $\hat{\mathbf{p}}$, as estimated by recursive least squares with exponential forgetting. The control to be applied at time t is then given by

$$u(t) = \frac{1}{\hat{b}_1} [y_r(t+1) + \hat{a}_1 y(t) + \hat{a}_2 y(t-1) + \dots + \hat{a}_{n_a} y(t+1-n_a) - \hat{b}_2 u(t-1) - \dots - \hat{b}_{n_b} \mathrm{u}(t+1-n_b)].$$

Such a simplistic control raises three problems.

— The estimated parameter $\hat{b}_1$ must differ sufficiently from zero for the computed controls to remain feasible. If $\hat{b}_1$ is zero, the control to be computed has no predictable influence on the output to be controlled, possibly because the delay between the input and output is larger than one time unit. One should then try a k-step-ahead predictor ($k > 1$).

— The output $y(t)$ is used to compute the control $u(t)$. The required computation must therefore be performed in a negligible time compared with the time constants of the process. Otherwise, the delay introduced should be taken into account during the analysis of the problem, and the prediction of the output at time $t + 1$ should not depend on $y(t)$ but satisfy

$$y_{mp}(t+1, \mathbf{p}) = f(\mathrm{u}(t), u(t-1), \dots, y(t-1), y(t-2), \dots, \mathbf{p}).$$

The control $u(t)$ can then be computed as a function of past measurements and the estimated parameters.

— This short-sighted policy amounts to approximately cancelling the zeros of the process by the poles of the controller. This will result in a loss of stability if the zeros of the process are outside the unit circle, which corresponds to the system having a non-minimal phase. Generalized predictive control avoids this difficulty.

To make the time between the measurement of $y(t+1)$ and the application of $u(t+1)$ as short as possible, one should compute, before time $t + 1$,

— the correction gain $\mathbf{k}(t+1)$,
— the matrix $\mathbf{P}(t+1)$ characterizing the uncertainty in the estimates,
— the prediction of the output $y_m[t+1, \hat{\mathbf{p}}_{ls}(t)]$,
— the next value $y_r(t+2)$ of the reference trajectory.

Starting at $t + 1$, one should

— measure $y(t+1)$ from the process,
— update the estimated parameters $\hat{\mathbf{p}}_{ls}(t+1)$,
— compute the input $u(t+1)$,
— apply it to the process.

The past is thus used to estimate the parameters, and the future to compute the control law. When there are no constraints on the input, $y_m[t+2, \hat{\mathbf{p}}_{ls}(t+1)]$ will be equal to $y_r(t+2)$ since the control $u(t+1)$ has been computed for precisely that purpose. Otherwise, the actual control applied to the process must be used to predict its output.

REMARK 4.4

Updating $\mathbf{P}$ at each iteration may turn out to take more time than is available. At the cost of the loss of the information contained in $\mathbf{P}$ and deterioration of the performance of the estimator, one may then employ parameter-updating algorithms with the same basic structure but simplified correction gains. One may, for instance, use

$$\hat{\mathbf{p}}(t+1) = \hat{\mathbf{p}}(t) + c\,\frac{\mathbf{r}(t)}{\mathbf{r}^{\mathrm{T}}(t)\mathbf{r}(t)}\,[y(t+1) - \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}(t)],$$

with $0 < c < 2$ (Richalet, Rault and Pouliquen, 1971; Richalet, 1991). As for recursive least squares with exponential forgetting, the correction gain of this algorithm never tends to zero. When the regressor vector $\mathbf{r}(t)$ is zero, the associated measurement contains no information on the parameters, which should not be updated. This can be implemented by adding a positive constant to the denominator of the correction gain (Kaczmarz, 1937).

One may alternatively use a *stochastic gradient* algorithm, to be considered again in Section 4.3.8),

$$\hat{\mathbf{p}}(t+1) = \hat{\mathbf{p}}(t) + \mathrm{c}(t)\mathbf{r}(t)[y(t+1) - \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}(t)],$$

where $c(t)$ is such that

$$c(t) > 0,\ \sum_{t=0}^{\infty} c(t) = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} c^2(t) < \infty.$$

One may choose, *e.g.*, $c(t) = c_1/t^{\alpha}$, where $0.5 < \alpha \leq 1$. One then gets a decreasing-gain algorithm (as in recursive least squares without forgetting) and the value of $c(t)$ must be increased when the prediction error exceeds a given threshold. In practice, the value of c_1 required for satisfactory performance is very much problem-dependent. Suitable gain-normalization techniques can eliminate this drawback (Section 6.4.3.2). See also the averaging technique described in Section 4.3.8. ◊

The use of data-recursive least squares to estimate the state of a system and possibly the parameters of its model leads naturally to the Kalman filter (Section 4.1.6).

To describe the behaviour of a given process, one sometimes hesitates between several LP model structures nested in the sense that the regressor vectors of the more complex structures incorporate those of the simpler ones. Comparing the performances of these structures may involve least-squares estimation of the parameters of each (Section 3.4). The technique described in the next paragraph makes it possible to compute these estimates recursively with respect to the number of parameters.

4.1.5 Parameter-recursive least squares

When the model structure is not imposed by prior considerations, one may wish to increase the number of parameters progressively until a satisfactory result is obtained. This can be done recursively, somewhat similarly to data-recursive least squares. Let $\hat{\mathbf{p}}_{i,k}$ be the vector consisting of the first i entries of $\hat{\mathbf{p}}$ when $\hat{\mathbf{p}}$ comprises k entries. Let $\mathbf{R}_k$ be the matrix of all the regressor vectors for the model with k parameters. The unweighted least-squares estimates for the models with k and $k+1$ parameters can then be written as

$$\hat{\mathbf{p}}_{k,k} = (\mathbf{R}_k^{\mathrm{T}}\mathbf{R}_k)^{-1}\mathbf{R}_k^{\mathrm{T}}\mathbf{y}^{\mathrm{s}},$$

and

$$\hat{\mathbf{p}}_{k+1,k+1} = (\mathbf{R}_{k+1}^{\mathrm{T}}\mathbf{R}_{k+1})^{-1}\mathbf{R}_{k+1}^{\mathrm{T}}\mathbf{y}^{\mathrm{s}} = \begin{bmatrix} \hat{\mathbf{p}}_{k,k+1} \\ \hat{p}_{+1,k+1} \end{bmatrix}.$$

It is assumed that the regressor matrix has a nested structure

$$\mathbf{R}_{k+1} = [\mathbf{R}_k \quad \mathbf{r}_{+1}].$$

Note that the dimension of the vector $\mathbf{r}_{+1}$ equals the number of data points, whereas the dimension of the regressor vector equals the number of parameters. The estimate with $k+1$ parameters satisfies

$$\begin{bmatrix} \mathbf{R}_k^{\mathrm{T}} \\ \mathbf{r}_{+1}^{\mathrm{T}} \end{bmatrix} [\mathbf{R}_k \quad \mathbf{r}_{+1}] \begin{bmatrix} \hat{\mathbf{p}}_{k,k+1} \\ \hat{p}_{+1,k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_k^{\mathrm{T}} \\ \mathbf{r}_{+1}^{\mathrm{T}} \end{bmatrix} \mathbf{y}^{\mathrm{s}} = \begin{bmatrix} \mathbf{R}_k^{\mathrm{T}}\mathbf{R}_k\hat{\mathbf{p}}_{k,k} \\ \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{y}^{\mathrm{s}} \end{bmatrix},$$

or equivalently

$$\mathbf{R}_k^{\mathrm{T}}\mathbf{R}_k\hat{\mathbf{p}}_{k,k+1} + \mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1}\hat{p}_{+1,k+1} = \mathbf{R}_k^{\mathrm{T}}\mathbf{R}_k\hat{\mathbf{p}}_{k,k},$$

$$\mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\hat{\mathbf{p}}_{k,k+1} + \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{r}_{+1}\hat{p}_{+1,k+1} = \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{y}^{\mathrm{s}}.$$

Subtracting $\mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\hat{\mathbf{p}}_{k,k}$ from both sides of the last equation, one can write

$$\begin{bmatrix} \mathbf{R}_k^{\mathrm{T}}\mathbf{R}_k & \mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1} \\ \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k & \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{r}_{+1} \end{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{p}}_{k,k+1} \\ \hat{p}_{+1,k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_{+1}^{\mathrm{T}}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}_k\hat{\mathbf{p}}_{k,k}) \end{bmatrix},$$

where $\delta\hat{\mathbf{p}}_{k,k+1}$ is the variation of the first k parameters due to adjoining the (k+1)th parameter, *i.e.*

$$\delta\hat{\mathbf{p}}_{k,k+1} = \hat{\mathbf{p}}_{k,k+1} - \hat{\mathbf{p}}_{k,k}.$$

This linear system of equations can be solved with the help of the relation

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1}[\mathbf{I} + \mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}] & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix},$$

which holds true provided that all matrices to be inverted are invertible. By setting

$$\mathbf{A}^{-1} = \mathbf{P}_k = (\mathbf{R}_k^{\mathrm{T}}\mathbf{R}_k)^{-1},\ \ \mathbf{B} = \mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1},\ \ \mathbf{C} = \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\ \text{ and }\ \mathbf{D} = \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{r}_{+1},$$

one gets

$$\mathbf{P}_{k+1} = \begin{bmatrix} \mathbf{P}_k + \dfrac{\mathbf{P}_k\mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1}\mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\mathbf{P}_k}{\mathbf{r}_{+1}^{\mathrm{T}}\mathbf{r}_{+1} - \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\mathbf{P}_k\mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1}} & -\dfrac{\mathbf{P}_k\mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1}}{\mathbf{r}_{+1}^{\mathrm{T}}\mathbf{r}_{+1} - \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\mathbf{P}_k\mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1}} \\ -\dfrac{\mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\mathbf{P}_k}{\mathbf{r}_{+1}^{\mathrm{T}}\mathbf{r}_{+1} - \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\mathbf{P}_k\mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1}} & \dfrac{1}{\mathbf{r}_{+1}^{\mathrm{T}}\mathbf{r}_{+1} - \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\mathbf{P}_k\mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1}} \end{bmatrix},$$

which implies

$$\begin{bmatrix} \delta\hat{\mathbf{p}}_{k,k+1} \\ \hat{p}_{+1,k+1} \end{bmatrix} = \mathbf{P}_{k+1} \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_{+1}^{\mathrm{T}}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}_k\hat{\mathbf{p}}_{k,k}) \end{bmatrix} = \begin{bmatrix} -\dfrac{\mathbf{P}_k\mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1}\mathbf{r}_{+1}^{\mathrm{T}}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}_k\hat{\mathbf{p}}_{k,k})}{\mathbf{r}_{+1}^{\mathrm{T}}\mathbf{r}_{+1} - \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\mathbf{P}_k\mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1}} \\ \dfrac{\mathbf{r}_{+1}^{\mathrm{T}}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}_k\hat{\mathbf{p}}_{k,k})}{\mathbf{r}_{+1}^{\mathrm{T}}\mathbf{r}_{+1} - \mathbf{r}_{+1}^{\mathrm{T}}\mathbf{R}_k\mathbf{P}_k\mathbf{R}_k^{\mathrm{T}}\mathbf{r}_{+1}} \end{bmatrix}.$$

As in data recursion, matrix inversion has been replaced by division by a scalar. If the counterpart $(\mathbf{y}^{\mathrm{s}} - \mathbf{R}_k\hat{\mathbf{p}}_{k,k})$ to the prediction error is zero, then the newly introduced parameter takes the value zero, and the first k parameters remain unchanged. For least-squares algorithms that are both data- and parameter-recursive, see Strobach (1991) and Orfanidis (1988), who provides the code of subroutines implementing them in an appendix.

4.1.5 Kalman filter

Consider a process described by

$$\mathbf{x}(t+1) = \mathbf{A}^*\mathbf{x}(t) + \mathbf{B}^*\mathbf{u}(t),$$

$$\mathbf{y}(t) = \mathbf{C}^*\mathbf{x}(t),$$

the state **x** of which is to be estimated from knowledge of its input-output behaviour. When $\mathbf{A}^*$, $\mathbf{B}^*$ and $\mathbf{C}^*$ are assumed known, the first idea that comes to mind is to build a mathematical model of the process driven by **u** and use the state of this model as an estimate for **x** (Figure 4.4).

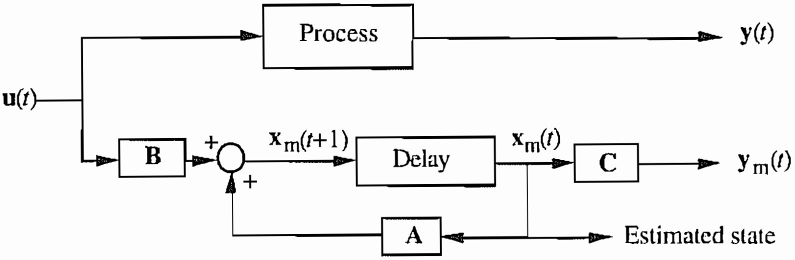


Figure 4.4. Naive observer

If this model were exact and stable, one could thus accurately estimate the state of the process, possibly after a transient period during which the effect of erroneous initial conditions might be felt. Such an open-loop solution might actually perform rather badly, because the model used is certainly not perfect. The basic idea of Kalman-Luenberger observers is to feed back the deviation $\mathbf{y} - \mathbf{y}_m$ to correct the state of the model, after passing it through a matrix gain **K** (Figure 4.5). The value of **K** dictates the dynamics of the observer, *i.e.* the speed with which its state will approach that of the process, provided that the model is correct and there is no noise.

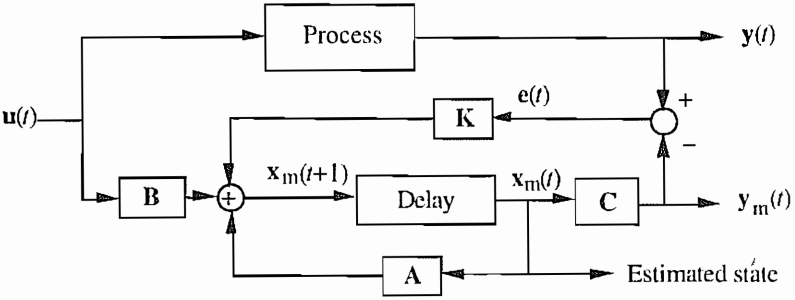


Figure 4.5. Kalman-Luenberger observer

The Kalman filter follows an analogous scheme, but explicitly takes into account the perturbations acting on the process and measurement noise. It extends without difficulty to time-varying systems, so assume that the system is described by

$$\mathbf{x}_{t+1} = \mathbf{A}_t\mathbf{x}_t + \mathbf{B}_t\mathbf{u}_t + \mathbf{v}_t,$$

$$\mathbf{y}_t = \mathbf{C}_t\mathbf{x}_t + \mathbf{w}_t.$$

The objective is to estimate the present state $\mathbf{x}_t$ of this system from the available information, which includes knowledge of the past values of the input $\mathbf{u}$ and output $\mathbf{y}$. The matrices $\mathbf{A}_t$, $\mathbf{B}_t$ and $\mathbf{C}_t$ are assumed to be *known* for all t. The process noise $\mathbf{v}_t$ corresponds to non-deterministic inputs, such as modelling errors, imperfections of actuators and external disturbances. The measurement noise $\mathbf{w}_t$ expresses the limitations of the sensors. These two noises are assumed to be zero-mean ($\mathrm{E}\{\mathbf{v}_t\} = \mathbf{0}$ and $\mathrm{E}\{\mathbf{w}_t\} = \mathbf{0}$) and to correspond to uncorrelated random vectors, such that

$$\mathrm{E}[\mathbf{v}_t\mathbf{w}_k^{\mathrm{T}}] = \mathbf{0}, \quad \mathrm{E}\{\mathbf{v}_t\mathbf{v}_k^{\mathrm{T}}\} = \mathbf{V}_t\delta_{tk}, \quad \mathrm{E}\{\mathbf{w}_t\mathbf{w}_k^{\mathrm{T}}\} = \mathbf{W}_t\delta_{tk},$$

where $\mathbf{V}_t$ and $\mathbf{W}_t$ are *known* positive-definite symmetric matrices and δ_{tk} is one if $t = k$ and zero otherwise. The larger these matrices are, the more erratic the behaviour of the system becomes; $\mathbf{V}_t$ attributes this to process noise, whereas $\mathbf{W}_t$ attributes it to errors in measurements. The initial state $\mathbf{x}_0$ of the system is taken as random, with *known* mean $\mathbf{m}_0$ and covariance $\mathbf{X}_0$. Moreover, $\mathbf{x}_0$ and $\mathbf{v}$ are assumed to be uncorrelated.

With no pretension to mathematical rigour (as can be found, *e.g.*, in (Anderson and Moore, 1979; Caines, 1988)), this section will show how the equations of the Kalman filter can be derived very simply from those of recursive least squares. It will proceed in three steps:

— the extension of recursive least squares to vector outputs,
— the study of a static system with no process noise,
— the study of a dynamic system with process noise.

The problems raised by implementation of the Kalman filter and its extension to parameter estimation and stochastic identification will then be considered.

REMARK 4.5

Some of the assumptions above can easily be replaced by less restrictive ones. It is possible to consider

— non-zero-mean $\mathbf{v}$ or $\mathbf{w}$,
— mutually correlated $\mathbf{v}$ and $\mathbf{w}$,
— measurement noise with a singular covariance matrix $\mathbf{W}_t$,
— coloured (autocorrelated) $\mathbf{v}$ and $\mathbf{w}$.

See, *e.g.*, (Borrie, 1992; Jazwinski, 1970). ◊

4.1.6.1 Vector data-recursive least squares

Consider a system for which the vector $\mathbf{y}_{t+1}$ consists of n_{y} scalar outputs. With this system is associated an LP model structure described by

$$\mathbf{y}_{\mathrm{m}}(t+1, \mathbf{p}) = \mathbf{R}_t^{\mathrm{T}}\mathbf{p},$$

where the regressor $\mathbf{R}_t$ is now an $n_{\rm p} \times n_{\rm y}$ matrix. The weighted least-squares estimate $\hat{\mathbf{p}}_{\rm ls}$ minimizes the cost

$$j_{\rm ls}(\mathbf{p}) = \sum_{t=1}^{n_t} (\mathbf{y}_t - \mathbf{R}_{t-1}^{\rm T}\mathbf{p})^{\rm T}\mathbf{Q}_t(\mathbf{y}_t - \mathbf{R}_{t-1}^{\rm T}\mathbf{p}),$$

where $\mathbf{Q}_t$ is some symmetric weighting matrix, assumed here to be positive-definite. It can be computed recursively by the following algorithm, similar to that established in the scalar case:

$$\mathbf{K}_{t+1} = \mathbf{P}_t\mathbf{R}_t(\mathbf{Q}_{t+1}^{-1} + \mathbf{R}_t^{\rm T}\mathbf{P}_t\mathbf{R}_t)^{-1},$$

$$\mathbf{P}_{t+1} = \mathbf{P}_t - \mathbf{K}_{t+1}\mathbf{R}_t^{\rm T}\mathbf{P}_t,$$

$$\mathbf{y}_{\rm m}[t+1, \hat{\mathbf{p}}_{\rm ls}(t)] = \mathbf{R}_t^{\rm T}\hat{\mathbf{p}}_{\rm ls}(t),$$

$$\hat{\mathbf{p}}_{\rm ls}(t+1) = \hat{\mathbf{p}}_{\rm ls}(t) + \mathbf{K}_{t+1}[\mathbf{y}_{t+1} - \mathbf{y}_{\rm m}[t+1, \hat{\mathbf{p}}_{\rm ls}(t)]].$$

Even when $\mathbf{Q}_{t+1}^{-1}$ is available *a priori*, each iteration requires the inversion of an $n_{\rm y} \times n_{\rm y}$ matrix. $\mathbf{P}_t$ can be interpreted as the covariance matrix of the estimation error $\mathrm{E}\{[\mathbf{p} - \hat{\mathbf{p}}_{\rm ls}(t)][\mathbf{p} - \hat{\mathbf{p}}_{\rm ls}(t)]^{\rm T}\}$, provided that $\mathbf{Q}_t$ is the inverse of the covariance of the measurement noise in $\mathbf{y}_t$ (Chapter 5).

4.1.6.2 Static system without process noise

When the state of the system is assumed to be constant, the state and observation equations reduce to

$$\mathbf{x}_{t+1} = \mathbf{x}_t,$$

$$\mathbf{y}_t = \mathbf{C}_t\mathbf{x}_t + \mathbf{w}_t.$$

Given the assumptions on $\mathbf{w}$, it seems natural to estimate $\mathbf{x}$ by minimizing a cost quadratic in the output error, weighted by the inverse of the covariance of the measurement noise. This corresponds to a Gauss-Markov estimator, which would become a maximum-likelihood estimator if $\mathbf{w}$ were additionally assumed to be Gaussian.

A direct application of the least-squares algorithm with $\hat{\mathbf{p}}_{\rm ls}(t) = \hat{\mathbf{x}}_t$, $\mathbf{Q}_t = \mathbf{W}_t^{-1}$ and $\mathbf{R}_t^{\rm T} = \mathbf{C}_{t+1}$ then leads to

$$\mathbf{K}_{t+1} = \mathbf{P}_t\mathbf{C}_{t+1}^{\rm T}(\mathbf{W}_{t+1} + \mathbf{C}_{t+1}\mathbf{P}_t\mathbf{C}_{t+1}^{\rm T})^{-1},$$

$$\mathbf{P}_{t+1} = \mathbf{P}_t - \mathbf{K}_{t+1}\mathbf{C}_{t+1}\mathbf{P}_t,$$

$$\mathbf{y}_{\rm m}(t+1, \hat{\mathbf{x}}_t) = \mathbf{C}_{t+1}\hat{\mathbf{x}}_t,$$

$$\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t + \mathbf{K}_{t+1}[\mathbf{y}_{t+1} - \mathbf{y}_{\rm m}(t+1, \hat{\mathbf{x}}_t)].$$

These equations are those of the Kalman filter. $\mathbf{P}_t$ is the covariance matrix $\mathrm{E}\{(\mathbf{x}_t - \hat{\mathbf{x}}_t)(\mathbf{x}_t - \hat{\mathbf{x}}_t)^{\rm T}\}$ of the estimation error.

4.1.6.3 Dynamic system with process noise

Consider now the initial system

$$\mathbf{x}_{t+1} = \mathbf{A}_t\mathbf{x}_t + \mathbf{B}_t\mathbf{u}_t + \mathbf{v}_t,$$

$$\mathbf{y}_t = \mathbf{C}_t\mathbf{x}_t + \mathbf{w}_t.$$

Its state evolves between measurement times, because of its dynamics, and the process noise $\mathbf{v}$ makes this evolution uncertain. This leads us to distinguish:

- the *predictions* of $\hat{\mathbf{x}}$ and $\mathbf{P}$ at time $t + 1$ given the information available at time t, denoted by $\hat{\mathbf{x}}_{t+1|t}$ and $\mathbf{P}_{t+1|t}$ and called *prior values*;
- the *updated* values of $\hat{\mathbf{x}}$ and $\mathbf{P}$ at time $t + 1$ obtained by taking the measurements collected at time $t + 1$ into account, denoted by $\hat{\mathbf{x}}_{t+1|t+1}$ and $\mathbf{P}_{t+1|t+1}$ and called *posterior values*.

Predicting $\hat{\mathbf{x}}$ *and* $\mathbf{P}$. The process noise $\mathbf{v}$ belongs to a sequence of independent random vectors. The past observations therefore bring no information on its present value. By replacing $\mathbf{v}_t$ by its mean value, which is zero, one gets the predictor

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{A}_t\hat{\mathbf{x}}_{t|t} + \mathbf{B}_t\mathbf{u}_t.$$

This prediction corresponds to the noise-free evolution that the system would have if its initial condition were $\hat{\mathbf{x}}_{t|t}$. The prediction error is then

$$\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t} = \mathbf{A}_t(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}) + \mathbf{v}_t.$$

If the prediction is unbiased, *i.e.* if $\mathrm{E}\{\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}\} = \mathbf{0}$ (which will hold true if $\mathrm{E}\{\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}\} = \mathbf{0}$), then the covariance of the prediction error is given by

$$\mathbf{P}_{t+1|t} = \mathrm{E}\{(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t})(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t})^{\mathrm{T}}\}$$

$$= \mathrm{E}\{\mathbf{A}_t(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})^{\mathrm{T}}\mathbf{A}_t^{\mathrm{T}}\} + \mathrm{E}\{\mathbf{v}_t\mathbf{v}_t^{\mathrm{T}}\}$$

$$= \mathbf{A}_t\mathbf{P}_{t|t}\mathbf{A}_t^{\mathrm{T}} + \mathbf{V}_t.$$

All other terms are zero, for $\mathbf{v}_t$ and $\mathbf{x}_t - \hat{\mathbf{x}}_{t|t}$ are mutually uncorrelated.

Updating $\hat{\mathbf{x}}$ *and* $\mathbf{P}$. During this phase, the knowledge of $\mathbf{x}_{t+1}$ and $\mathbf{P}_{t+1}$ is improved by substituting $\hat{\mathbf{x}}_{t+1|t+1}$ for $\hat{\mathbf{x}}_{t+1|t}$ and $\mathbf{P}_{t+1|t+1}$ for $\mathbf{P}_{t+1|t}$, so as to take the results of measurements at time $t + 1$ into account. This corresponds to one step of a *static* problem, since the actual value of $\mathbf{x}_{t+1}$ does not change during the updating of its estimate. The equations of the static case can therefore be employed, provided that the following notation changes are made, to distinguish the prior and posterior values of $\hat{\mathbf{x}}_{t+1}$ and $\mathbf{P}_{t+1}$:

$$\hat{\mathbf{x}}_{t+1} \rightarrow \hat{\mathbf{x}}_{t+1|t+1},$$
$$\hat{\mathbf{x}}_{t} \rightarrow \hat{\mathbf{x}}_{t+1|t},$$
$$\mathbf{P}_{t+1} \rightarrow \mathbf{P}_{t+1|t+1},$$
$$\mathbf{P}_{t} \rightarrow \mathbf{P}_{t+1|t}.$$

Assume that $\hat{\mathbf{x}}_{t|t}$ and $\mathbf{P}_{t|t}$ are available (initialization will be considered later). An iteration of the algorithm corresponds to getting $\hat{\mathbf{x}}_{t+1|t+1}$ and $\mathbf{P}_{t+1|t+1}$, *i.e.* to

— predicting the state and the covariance of the prediction error

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{A}_t\hat{\mathbf{x}}_{t|t} + \mathbf{B}_t\mathbf{u}_t,$$

$$\mathbf{P}_{t+1|t} = \mathbf{A}_t\mathbf{P}_{t|t}\mathbf{A}_t^{\mathrm{T}} + \mathbf{V}_t;$$

— computing the gain of the filter

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{C}_{t+1}^{\mathrm{T}}(\mathbf{W}_{t+1} + \mathbf{C}_{t+1}\mathbf{P}_{t+1|t}\mathbf{C}_{t+1}^{\mathrm{T}})^{-1};$$

— updating the state estimate

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + \mathbf{K}_{t+1}(\mathbf{y}_{t+1} - \mathbf{C}_{t+1}\hat{\mathbf{x}}_{t+1|t});$$

— updating the covariance estimate

$$\mathbf{P}_{t+1|t+1} = \mathbf{P}_{t+1|t} - \mathbf{K}_{t+1}\mathbf{C}_{t+1}\mathbf{P}_{t+1|t}.$$

As with recursive least squares without forgetting, it can be checked that

$$\mathbf{P}_{t+1|t+1} \leq \mathbf{P}_{t+1|t}.$$

The information provided by $\mathbf{y}_{t+1}$ can therefore only decrease the uncertainty in the state.

REMARKS 4.6

— Computing $\mathbf{K}_{t+1}$ requires inverting $(\mathbf{W}_{t+1} + \mathbf{C}_{t+1}\mathbf{P}_{t+1|t}\mathbf{C}_{t+1}^{\mathrm{T}})$, always possible if $\mathbf{W}_{t+1}$ is invertible. It is paradoxically when there is no measurement noise that difficulties requiring specific treatment appear.

— This is an *estimating filter*, which computes $\hat{\mathbf{x}}_{t+1|t+1}$ from $\hat{\mathbf{x}}_{t|t}$. It can be transformed into a *predicting filter*, computing $\hat{\mathbf{x}}_{t+1|t}$ from $\hat{\mathbf{x}}_{t|t-1}$. Since

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{A}_t\hat{\mathbf{x}}_{t|t} + \mathbf{B}_t\mathbf{u}_t \quad \text{and} \quad \hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{C}_t\hat{\mathbf{x}}_{t|t-1}),$$

this predicting filter is given by

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{A}_t\hat{\mathbf{x}}_{t|t-1} + \mathbf{B}_t\mathbf{u}_t + \tilde{\mathbf{K}}_t(\mathbf{y}_t - \mathbf{C}_t\hat{\mathbf{x}}_{t|t-1}),$$

with $\tilde{\mathbf{K}}_t = \mathbf{A}_t\mathbf{K}_t$.

- The special case of a static system with process noise, which corresponds to $\mathbf{A}_t \equiv \mathbf{I}$ and $\mathbf{B}_t \equiv \mathbf{0}$, provides an alternative method for tracking time-varying parameters to the exponential forgetting of Section 4.1.4.2. The parameters then play the role of the state and the process noise is made responsible for their fluctuations. $\mathbf{C}_t$ consists of regressors and may depend on measurements up to time $t-1$. This approach is of special interest when information is available on the possible variations of each parameter. The choice of the matrix $\mathbf{V}_t$ gives more degrees of freedom than that of the scalar forgetting factor λ. One may, for instance, take into account the fact that some components of $\mathbf{p}^*$ are assumed to be constant. ◊

A most important feature of the Kalman filter is its recursive nature, which makes its on-line implementation particularly easy. To reduce real-time computation, one may compute off-line all quantities that do not depend on the measurements.

4.1.6.4 Off-line computation

Initialization. If the mean $\mathbf{m}_0$ and covariance $\mathbf{X}_0$ of the initial state $\mathbf{x}_0$ are known, one should choose $\hat{\mathbf{x}}_{0|0} = \mathbf{m}_0$ and $\mathbf{P}_{0|0} = \mathbf{X}_0$. By analogy with recursive least squares, one might otherwise choose $\hat{\mathbf{x}}_{0|0} = \mathbf{0}$ and $\mathbf{P}_{0|0} = c\mathbf{I}$, with c a large positive scalar, to express one's lack of confidence in the estimate of the initial state.

Iteration. Unless part of the model is not known in advance (as would be so, for instance, if the filter were used to estimate parameters and $\mathbf{C}_t$ incorporated observations, see the last point of Remarks 4.6), all correction gains and prior and posterior covariance matrices of the state estimation error can be computed off-line. Before any measurement, it is thus possible to assess what confidence can be placed in future state estimates, provided, of course, that the assumptions on which the Kalman filter relies are satisfied. This is both an advantage and a drawback:

- an *advantage*, for on-line computation will be much reduced,
- a *drawback*, for the filter operates in open loop. If the information used to compute it, *i.e.* the equations of the model and noise characteristics, is too inaccurate, it may then diverge, with the discrepancy between the observed and predicted outputs increasing aberrantly. As with recursive least squares, a layer external to the algorithm proper should therefore be introduced to monitor the behaviour of the filter through the evolution of the output-prediction error (Section 4.1.6.7).

4.1.6.5 On-line computation

If the sequence of correction gains $\mathbf{K}_t$ has been computed beforehand, on-line computation reduces to predicting $\hat{\mathbf{x}}_{t+1|t}$ and updating it to $\hat{\mathbf{x}}_{t+1|t+1}$.

4.1.6.6 Influence of the covariances of the process and measurement noise

The formula used to compute $\mathbf{K}_{t+1}$ implies that

$$\mathbf{K}_{t+1}(\mathbf{W}_{t+1} + \mathbf{C}_{t+1}\mathbf{P}_{t+1|t}\mathbf{C}_{t+1}^{\mathrm{T}}) = \mathbf{P}_{t+1|t}\mathbf{C}_{t+1}^{\mathrm{T}}.$$

After right multiplication by $\mathbf{W}_{t+1}^{-1}$, one gets

$$\mathbf{K}_{t+1}(\mathbf{I} + \mathbf{C}_{t+1}\mathbf{P}_{t+1|t}\mathbf{C}_{t+1}^{\mathrm{T}}\mathbf{W}_{t+1}^{-1}) = \mathbf{P}_{t+1|t}\mathbf{C}_{t+1}^{\mathrm{T}}\mathbf{W}_{t+1}^{-1},$$

so

$$\mathbf{K}_{t+1} = (\mathbf{P}_{t+1|t} - \mathbf{K}_{t+1}\mathbf{C}_{t+1}\mathbf{P}_{t+1|t})\mathbf{C}_{t+1}^{\mathrm{T}}\mathbf{W}_{t+1}^{-1}.$$

Taking the expression for $\mathbf{P}_{t+1|t+1}$ into account, one can therefore write

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t+1}\mathbf{C}_{t+1}^{\mathrm{T}}\mathbf{W}_{t+1}^{-1}.$$

For any given $\mathbf{W}_{t+1}$, the gain of the filter increases with $\mathbf{P}_{t+1|t+1}$. Now $\mathbf{P}_{t+1|t+1}$ will be large if $\mathbf{P}_{0|0}$ is (but this initial effect quickly disappears), and more importantly if the process noise is large, *i.e.* if $\mathbf{V}_t$ is large.

The larger the correction gain is, the more jittery the behaviour of the filter will be, *i.e.* the more drastically the state estimate will be modified to take new measurements into account. This behaviour expresses a lack of confidence in the predicted state estimate.

Conversely, for any given $\mathbf{P}_{t+1|t+1}$, the larger the covariance of the measurement noise $\mathbf{W}_{t+1}$ is, the smaller the correction gain will be and the less the new measurements will be taken into account to update the state estimate. This expresses a lack of confidence in the new measurements.

4.1.6.7 Detection of divergence

One can test whether the filter is operating correctly by computing the covariance of the deviation between the measured and predicted outputs, so as to detect any very improbable deviations. Let $\tilde{\mathbf{y}}_{t+1}$ be the output-prediction error

$$\tilde{\mathbf{y}}_{t+1} = \mathbf{y}_{t+1} - \hat{\mathbf{y}}_{t+1|t} = \mathbf{C}_{t+1}(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1|t}) + \mathbf{w}_{t+1}.$$

If $\mathrm{E}\{\tilde{\mathbf{y}}_{t+1}\} = \mathbf{0}$, which holds true if the initial state estimate is unbiased, then the covariance of the output-prediction error is

$$\mathrm{E}\{\tilde{\mathbf{y}}_{t+1}\tilde{\mathbf{y}}_{t+1}^{\mathrm{T}}\} = \mathbf{C}_{t+1}\mathbf{P}_{t+1|t}\mathbf{C}_{t+1}^{\mathrm{T}} + \mathbf{W}_{t+1},$$

and the standard deviation associated with each output is the square root of the corresponding diagonal entry. Any deviation between the predicted and measured outputs that exceeds three standard deviations can be considered very unlikely. Unless this is just a fleeting phenomenon, it indicates that the filter is diverging. Divergence may be due, for instance, to

— erroneous modelling of the dynamics of the process (badly chosen $\mathbf{A}_t$, $\mathbf{B}_t$ and $\mathbf{C}_t$, violation of the hypothesis of linearity);
— underestimation of the process noise, with $\mathbf{V}_t$ too small;
— incorrect initialization, with $\mathbf{P}_{0|0}$ too small.

An *ad hoc* way to avoid divergence is to increase the covariance $\mathbf{V}_t$ of the process noise. Another approach is to try to estimate a parameter vector $\mathbf{p}$ consisting of all unknown elements of the model ($\mathbf{A}_t$, $\mathbf{B}_t$, $\mathbf{C}_t$, $\mathbf{V}_t$ and $\mathbf{W}_t$) from the available experimental data. One may, for instance, estimate $\mathbf{p}$ in the maximum-likelihood sense (see also Sections 4.1.6.10 and 4.1.6.11). If the state variables are meant to have a concrete meaning, one should make sure to select an input-output configuration and a parametrization that ensure global identifiability of the model structure.

REMARK 4.7

As with least squares, the details of the numerical implementation of the filter affect greatly the precision and stability of the results. We shall mention only two points.

— Bierman's U-D factorization algorithm (1977) can be used to keep the covariance matrices symmetric and positive-definite, which is crucial to ensure satisfactory operation.

— When the discrete-time model is associated with a continuous-time process and the sampling period is short compared with the time constants of the system, implementation using the δ operator is recommended (Middleton and Goodwin, 1990) ◊

4.1.6.8 Stationary filter

An important special case is when

— the system studied is time-invariant (**A**, **B** and **C** are constant),

— the statistics of the process and measurement noises are stationary (**V** and **W** are constant).

If the pair (**C**, **A**) is detectable, *i.e.* if any mode of **A** that is not observable *via* **C** is asymptotically stable, $\mathbf{P}_{t|t}$ tends to a constant value as t tends to infinity (Anderson and Moore, 1979). Let

$$\mathbf{P} = \lim_{t \to \infty} \mathbf{P}_{t|t}, \text{ and } \mathbf{P}^+ = \lim_{t \to \infty} \mathbf{P}_{t+1|t}.$$

From the equation for $\mathbf{P}_{t+1|t}$, one gets

$$\mathbf{P}^+ = \mathbf{APA}^\mathrm{T} + \mathbf{V}.$$

Taking the expressions for $\mathbf{K}_{t+1}$ and $\mathbf{P}_{t+1|t+1}$ into account, one can write

$$\mathbf{P} = \mathbf{P}^+ - \mathbf{P}^+\mathbf{C}^\mathrm{T}(\mathbf{W} + \mathbf{CP}^+\mathbf{C}^\mathrm{T})^{-1}\mathbf{CP}^+.$$

Multiplying this last equation by **A** on the left and by $\mathbf{A}^\mathrm{T}$ on the right and replacing $\mathbf{APA}^\mathrm{T}$ in the result by $\mathbf{P}^+ - \mathbf{V}$ gives the discrete Riccati equation

$$\mathbf{P}^+ - \mathbf{V} = \mathbf{AP}^+\mathbf{A}^\mathrm{T} - \mathbf{AP}^+\mathbf{C}^\mathrm{T}(\mathbf{W} + \mathbf{CP}^+\mathbf{C}^\mathrm{T})^{-1}\mathbf{CP}^+\mathbf{A}^\mathrm{T}.$$

The matrix $\mathbf{P}^+$ can be computed in two ways:

- as the positive-definite solution of this Riccati equation,
- by iterating the equations for the evolution of the covariance until $\mathbf{P}_{t+1|t}$ becomes constant.

The gain of the stationary Kalman filter is then simply given by

$$\mathbf{K} = \mathbf{P}^{+}\mathbf{C}^{\mathrm{T}}(\mathbf{W} + \mathbf{C}\mathbf{P}^{+}\mathbf{C}^{\mathrm{T}})^{-1}.$$

It depends on $\mathbf{A}$, $\mathbf{C}$, $\mathbf{V}$ and $\mathbf{W}$, but not on $\mathbf{P}_{0|0}$ or $\mathbf{B}$. The implementation of the stationary filter is particularly simple, since the prediction of the state and updating of its estimate can be combined into

$$\hat{\mathbf{x}}_{t+1|t+1} = \mathbf{A}\hat{\mathbf{x}}_{t|t} + \mathbf{B}\mathbf{u}_t + \mathbf{K}[\mathbf{y}_{t+1} - \mathbf{C}(\mathbf{A}\hat{\mathbf{x}}_{t|t} + \mathbf{B}\mathbf{u}_t)]$$

$$= (\mathbf{A} - \mathbf{K}\mathbf{C}\mathbf{A})\hat{\mathbf{x}}_{t|t} + (\mathbf{B} - \mathbf{K}\mathbf{C}\mathbf{B})\mathbf{u}_t + \mathbf{K}\mathbf{y}_{t+1}.$$

Note the similarity to the Kalman-Luenberger observer.

4.1.6.9 Use for the choice of sensors

To instrument a system in order to estimate its state, one should be able to answer the following questions.

- What are the critical quantities to be measured, *i.e.* what should the vector $\mathbf{y}$ consist of?
- What quality of measurements is required to ensure a given quality of state estimate? How should the sensors be selected, given their costs and precisions?

These questions can be addressed in the framework of experiment design (Chapter 6). The Kalman filter is a basic tool to answer them, since it computes the sequences of covariance matrices $\mathbf{P}_{t+1|t+1}$ and $\mathbf{P}_{t+1|t}$. To compare the performance of two configurations of sensors, one just has to solve the equations describing the evolution of the covariances of the corresponding estimation errors. The smaller the diagonal entries of the covariance matrices are, the more precise the state estimation will be, provided that the models describe the system and sensors correctly... These covariances can be computed off-line, without making any measurements on the system, which therefore need not be built.

4.1.6.10 Extended Kalman filter: real-time parameter estimation

Now consider a system described by the following possibly non-LI discrete-time model

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{p}_t) + \mathbf{v}_t,$$

$$\mathbf{x}_0 = \mathbf{x}_0(\mathbf{p}_0),$$

$$\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{p}_t) + \mathbf{w}_t.$$

This model depends on a vector of unknown parameters $\mathbf{p}_t$, possibly time-varying. It may result from discretization of a continuous-time model. One wishes to estimate $\mathbf{x}_t$ and $\mathbf{p}_t$ simultaneously, hence the idea of defining an extended state vector

$$\mathbf{x}_t^{\mathrm{e}} = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{p}_t \end{bmatrix}.$$

Provided that an evolution equation is chosen for the parameters, such as

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \mathbf{v}_t^{\mathrm{p}},$$

the evolution of the extended state can be written as

$$\mathbf{x}_{t+1}^{\mathrm{e}} = \begin{bmatrix} \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{p}_t) \\ \mathbf{p}_t \end{bmatrix} + \begin{bmatrix} \mathbf{v}_t \\ \mathbf{v}_t^{\mathrm{p}} \end{bmatrix} = \mathbf{f}^{\mathrm{e}}(\mathbf{x}_t^{\mathrm{e}}, \mathbf{u}_t) + \mathbf{v}_t^{\mathrm{e}},$$

$$\mathbf{x}_0^{\mathrm{e}} = \begin{bmatrix} \mathbf{x}_0(\mathbf{p}_0) \\ \mathbf{p}_0 \end{bmatrix}.$$

The observation equation becomes

$$\mathbf{y}_t = \mathbf{h}^{\mathrm{e}}(\mathbf{x}_t^{\mathrm{e}}) + \mathbf{w}_t.$$

Even when the initial model is LI, such is no longer the case for the extended model, so the Kalman filter does not apply directly. We shall only present here the first-order extended Kalman filter, and the reader may refer, *e.g.*, to (Bar-Shalom and Fortmann, 1988) for further details, including the second-order filter.

Replace $\mathbf{f}^{\mathrm{e}}(\mathbf{x}_t^{\mathrm{e}}, \mathbf{u}_t)$ by its first-order expansion around $\hat{\mathbf{x}}_{t|t}^{\mathrm{e}}$:

$$\mathbf{f}^{\mathrm{e}}(\mathbf{x}_t^{\mathrm{e}}, \mathbf{u}_t) \approx \mathbf{f}^{\mathrm{e}}(\hat{\mathbf{x}}_{t|t}^{\mathrm{e}}, \mathbf{u}_t) + \mathbf{A}_t(\mathbf{x}_t^{\mathrm{e}} - \hat{\mathbf{x}}_{t|t}^{\mathrm{e}}),$$

with

$$\mathbf{A}_t = \frac{\partial \mathbf{f}^{\mathrm{e}}(\mathbf{x}_t^{\mathrm{e}}, \mathbf{u}_t)}{\partial \mathbf{x}_t^{\mathrm{e}\mathrm{T}}}\Big|_{\hat{\mathbf{x}}_{t|t}^{\mathrm{e}}} = \begin{bmatrix} \dfrac{\partial \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{p}_t)}{\partial \mathbf{x}_t^{\mathrm{T}}} & \dfrac{\partial \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{p}_t)}{\partial \mathbf{p}_t^{\mathrm{T}}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\Big|_{\hat{\mathbf{x}}_{t|t}^{\mathrm{e}}},$$

and replace $\mathbf{h}^{\mathrm{e}}(\mathbf{x}_t^{\mathrm{e}})$ by its first-order expansion around $\hat{\mathbf{x}}_{t|t-1}^{\mathrm{e}}$:

$$\mathbf{h}^{\mathrm{e}}(\mathbf{x}_t^{\mathrm{e}}) \approx \mathbf{h}^{\mathrm{e}}(\hat{\mathbf{x}}_{t|t-1}^{\mathrm{e}}) + \mathbf{C}_t(\mathbf{x}_t^{\mathrm{e}} - \hat{\mathbf{x}}_{t|t-1}^{\mathrm{e}}),$$

with

$$\mathbf{C}_t = \frac{\partial \mathbf{h}^{\mathrm{e}}(\mathbf{x}_t^{\mathrm{e}})}{\partial \mathbf{x}_t^{\mathrm{e}\mathrm{T}}}\Big|_{\hat{\mathbf{x}}_{t|t-1}^{\mathrm{e}}} = \begin{bmatrix} \dfrac{\partial \mathbf{h}(\mathbf{x}_t, \mathbf{p}_t)}{\partial \mathbf{x}_t^{\mathrm{T}}} & \dfrac{\partial \mathbf{h}(\mathbf{x}_t, \mathbf{p}_t)}{\partial \mathbf{p}_t^{\mathrm{T}}} \end{bmatrix}\Big|_{\hat{\mathbf{x}}_{t|t-1}^{\mathrm{e}}}.$$

The non-LI state and observation equations are then approximated by

$$\mathbf{x}^{\mathrm{e}}_{t+1} = \mathbf{A}_t\mathbf{x}^{\mathrm{e}}_t + \mathbf{f}^{\mathrm{e}}(\hat{\mathbf{x}}^{\mathrm{e}}_{t|t}, \mathbf{u}_t) - \mathbf{A}_t\hat{\mathbf{x}}^{\mathrm{e}}_{t|t} + \mathbf{v}^{\mathrm{e}}_t,$$

$$\mathbf{y}_t = \mathbf{C}_t\mathbf{x}^{\mathrm{e}}_t + \mathbf{h}^{\mathrm{e}}(\hat{\mathbf{x}}^{\mathrm{e}}_{t|t-1}) - \mathbf{C}_t\hat{\mathbf{x}}^{\mathrm{e}}_{t|t-1} + \mathbf{w}_t.$$

Since this approximation is linear in $\mathbf{x}^{\mathrm{e}}_t$, the Kalman filtering equations can be used, provided that the statistics of $\mathbf{v}^{\mathrm{e}}_t$ and $\mathbf{w}_t$ are specified. Unless information is available to the contrary, it is usually assumed that

$$\mathrm{E}\{\mathbf{v}^{\mathrm{e}}_t\} = \mathbf{0},\ \mathrm{E}\{\mathbf{w}_t\} = \mathbf{0},\ \mathrm{E}\{\mathbf{v}^{\mathrm{e}}_t\mathbf{w}^{\mathrm{T}}_k\} = \mathbf{0},\ \mathrm{E}\{\mathbf{v}^{\mathrm{e}}_t\mathbf{v}^{\mathrm{e}\mathrm{T}}_k\} = \mathbf{V}^{\mathrm{e}}\delta_{tk},\ \mathrm{E}\{\mathbf{w}_t\mathbf{w}^{\mathrm{T}}_k\} = \mathbf{W}\delta_{tk},$$

with

$$\mathbf{V}^{\mathrm{e}} = \begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^{\mathrm{p}} \end{bmatrix}.$$

The matrices $\mathbf{V}$, $\mathbf{V}^{\mathrm{p}}$ and $\mathbf{W}$ are most often chosen to be diagonal. The larger the ith diagonal entry of $\mathbf{V}^{\mathrm{p}}$ (respectively $\mathbf{V}$) is, the more quickly the filter will modify the estimate of the ith component of $\mathbf{p}_t$ (respectively $\mathbf{x}_t$) in the light of the measurements. Conversely, the larger the ith diagonal entry of $\mathbf{W}$ is, the less the filter will take the measurements from the ith sensor into account. The designer of an extended Kalman filter will thus be able to play with these tuning parameters to search for a satisfactory compromise. In particular, it is possible to indicate that some parameter must tend towards a constant value by setting the corresponding diagonal entry of $\mathbf{V}^{\mathrm{p}}$ to zero. If the ith component of $\mathbf{x}$ or $\mathbf{p}$ is assumed to be liable to vary very quickly and unpredictably, this can be taken into account by setting the ith entry of $\mathbf{V}$ or $\mathbf{V}^{\mathrm{p}}$ to a large value.

Assume that $\hat{\mathbf{x}}^{\mathrm{e}}_{t|t}$ and $\mathbf{P}_{t|t}$ are available. An iteration then consists of computing $\hat{\mathbf{x}}^{\mathrm{e}}_{t+1|t+1}$ and $\mathbf{P}_{t+1|t+1}$. To predict the future state, the nonlinear state equation

$$\hat{\mathbf{x}}^{\mathrm{e}}_{t+1|t} = \mathbf{f}^{\mathrm{e}}(\hat{\mathbf{x}}^{\mathrm{e}}_{t|t}, \mathbf{u}_t)$$

can be used, rather than its linearized version. $\mathbf{P}_{t+1|t}$, $\mathbf{K}_{t+1}$ and $\mathbf{P}_{t+1|t+1}$ are given by the usual formulas

$$\mathbf{P}_{t+1|t} = \mathbf{A}_t\mathbf{P}_{t|t}\mathbf{A}^{\mathrm{T}}_t + \mathbf{V}^{\mathrm{e}},$$

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{C}^{\mathrm{T}}_{t+1}(\mathbf{W} + \mathbf{C}_{t+1}\mathbf{P}_{t+1|t}\mathbf{C}^{\mathrm{T}}_{t+1})^{-1},$$

$$\mathbf{P}_{t+1|t+1} = \mathbf{P}_{t+1|t} - \mathbf{K}_{t+1}\mathbf{C}_{t+1}\mathbf{P}_{t+1|t}.$$

To update the state estimate, the actual prediction error, as computed from the nonlinear observation equation, can be used rather than its linearized counterpart:

$$\hat{\mathbf{x}}^{\mathrm{e}}_{t+1|t+1} = \hat{\mathbf{x}}^{\mathrm{e}}_{t+1|t} + \mathbf{K}_{t+1}[\mathbf{y}_{t+1} - \mathbf{h}^{\mathrm{e}}(\hat{\mathbf{x}}^{\mathrm{e}}_{t+1|t})].$$

Provided that the filter converges, which is not guaranteed, one thus gets an estimate of $\mathbf{x}^{\mathrm{e}}_{t+1}$ which includes an estimate of the parameters. Heuristics are often introduced to force the linearized model to remain stable.

REMARK 4.8

The extended Kalman filter has been used in a large number of practical applications. Its popularity derives from its remarkable simplicity of implementation. It nevertheless presents some hard-to-comprehend divergence phenomena. In the special case of stationary LI models, a detailed analysis of the limitations of this approach has been proposed (Ljung, 1979). The introduction of a corrective term involving the sensitivity of the correction gain $\mathbf{K}_t$ to the parameters makes the convergence properties of the filter identical to those of the estimator obtained with a maximum-likelihood approach. ◊

4.1.6.11 Stochastic identification

Implementing a Kalman filter requires knowledge of $(\mathbf{A}_t, \mathbf{B}_t, \mathbf{C}_t, \mathbf{V}_t, \mathbf{W}_t)$ for all t. If this information is not readily available, it should be derived from the knowledge of the inputs and outputs, which is a stochastic identification problem. Consider the simple case where the system and noise are stationary and where a stationary Kalman filter is sought. A possible approach would then be to

— find a model $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{V}, \mathbf{W})$, by some method to be defined,
— solve the corresponding discrete Riccati equation to get the correction gain $\mathbf{K}$,
— implement the stationary Kalman filter thus obtained.

One may, however, drastically simplify the procedure by replacing the estimating filter by a predicting filter (Remarks 4.6), which can be written as

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{A}\hat{\mathbf{x}}_{t|t-1} + \mathbf{B}\mathbf{u}_t + \tilde{\mathbf{K}}(\mathbf{y}_t - \mathbf{C}\hat{\mathbf{x}}_{t|t-1}),$$

with $\tilde{\mathbf{K}} = \mathbf{AK}$. Let $\bar{\mathbf{y}}_t$ be the output-prediction error at time t (the *innovation*)

$$\bar{\mathbf{y}}_t = \mathbf{y}_t - \mathbf{C}\hat{\mathbf{x}}_{t|t-1}.$$

The last two equations can be rewritten as

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{A}\hat{\mathbf{x}}_{t|t-1} + \mathbf{B}\mathbf{u}_t + \tilde{\mathbf{K}}\bar{\mathbf{y}}_t,$$

$$\mathbf{y}_t = \mathbf{C}\hat{\mathbf{x}}_{t|t-1} + \bar{\mathbf{y}}_t.$$

One may even forget that $\hat{\mathbf{x}}_{t+1|t}$ is a state predictor, set $\mathbf{x}_t = \hat{\mathbf{x}}_{t|t-1}$ and model the process directly as

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \tilde{\mathbf{K}}\bar{\mathbf{y}}_t,$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \bar{\mathbf{y}}_t,$$

which is called the *innovations representation*. This form involves a single noise $\bar{\mathbf{y}}_t$, with the same dimension as the output, instead of two noises $\mathbf{v}_t$ and $\mathbf{w}_t$, with the dimensions of the state and output respectively. This presents two advantages (Ljung and Söderström, 1983):

— since the number of noise parameters is drastically reduced, modeling the stochastic process becomes much simpler;
— the model of the process includes $\widetilde{\mathbf{K}}$ explicitly. The correction gain of the stationary Kalman filter is therefore obtained directly, without having to solve any Riccati equation.

EXAMPLE 4.7

Consider the following canonical state-space representation, called the *companion form* or *observer form*, of a single-input-single-output time-invariant LI process:

$$\mathbf{x}_{t+1} = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & 0 & \dots \\ -a_3 & 0 & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ -a_n & 0 & \dots & \dots & 0 \end{bmatrix} \mathbf{x}_t + \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ b_n \end{bmatrix} u_t + \begin{bmatrix} \widetilde{k}_1 \\ \widetilde{k}_2 \\ . \\ . \\ \widetilde{k}_n \end{bmatrix} \widetilde{y}_t,$$

$$y_t = [\ 1 \quad 0 \quad \dots \quad 0 \quad 0\ ]\,\mathbf{x}_t + \widetilde{y}_t.$$

The $3n + 1$ parameters of this model are the a_i, b_i, $\widetilde{k}_i$ $(i = 1, \dots, n)$ and the variance of the innovation $\widetilde{y}_t$. The vector of the $\widetilde{k}_i$'s is the gain $\widetilde{\mathbf{k}}$ of the stationary Kalman filter. Define c_i as

$$c_i = \widetilde{k}_i + a_i\ ,\ i = 1, \dots, n.$$

The recurrence equation that corresponds to this canonical representation can then be written as

$$y_t + a_1 y_{t-1} + \dots + a_n y_{t-n} = b_1 u_{t-1} + \dots + b_n u_{t-n} + \widetilde{y}_t + c_1 \widetilde{y}_{t-1} + \dots + c_n \widetilde{y}_{t-n}.$$

It is therefore an ARMAX model (Section 2.4). Various techniques to estimate its parameters will be considered in the remainder of this chapter. This being done, the gain of the stationary Kalman filter is computed as $\widetilde{k}_i = c_i - a_i$. ◊

4.1.7 Errors-in-variables approach

As seen in Chapter 3, unweighted least squares correspond to maximum-likelihood estimation if the observations can be written as

$$y(t) = \mathbf{r}^{\mathrm{T}}(t-1)\mathbf{p}^* + \varepsilon(t),\ t = 1, \dots, n_t,$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, \sigma^2)$, or equivalently if

$$\mathbf{y}^{\mathrm{s}} = \mathbf{R}\mathbf{p}^* + \boldsymbol{\varepsilon},$$

where ε is distributed $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_{n_t})$. Only the output is then assumed to be noisy, whereas all variables involved in the computation of the regressor vector are assumed to be known exactly. This hypothesis (which Kalman (1982) calls *prejudice*) is often rather unrealistic, and one may prefer to assume that *all* variables involved in the estimation problem, *i.e.* $y(t)$ and all components of $\mathbf{r}(t-1)$, are noisy. This corresponds to the so-called *errors-in-variables* approach; see, *e.g.*, (Anderson, 1985). The output then loses its particular status of being the single noisy variable, and can be pooled with the regressor vector into a single vector

$$\mathbf{x}(t) = [y(t), \mathbf{r}^{\mathrm{T}}(t-1)]^{\mathrm{T}}.$$

If the vector $\mathbf{x}^*(t)$ of the corresponding *noise-free* variables satisfies a linear relation $\mathbf{x}^{*\mathrm{T}}\mathbf{p}^* = 0$, where the dimension of $\mathbf{p}^*$ has been increased by one to account for the incorporation of y in $\mathbf{x}$, then the matrix $\mathbf{X}^*$ obtained by piling up the vectors $\mathbf{x}^{*\mathrm{T}}(t)$ $(t = 0, \ldots, n_t)$ satisfies

$$\mathbf{X}^*\mathbf{p}^* = \mathbf{0} \Rightarrow \mathbf{X}^{*\mathrm{T}}\mathbf{X}^*\mathbf{p}^* = \mathbf{0}.$$

The true value $\mathbf{p}^*$ for the parameters of the linear relation therefore belongs to the kernel of the noise-free empirical covariance matrix

$$\boldsymbol{\Sigma}^* = \frac{1}{n_t}\mathbf{X}^{*\mathrm{T}}\,\mathbf{X}^*.$$

At best (provided that dim ker $\boldsymbol{\Sigma}^* = 1$), $\mathbf{p}^*$ is thus only identifiable up to a normalization coefficient. In the least-squares method, for instance, the chosen normalization policy associates a parameter equal to minus one with $y(t)$.

Unfortunately, one has no access to $\boldsymbol{\Sigma}^*$, but only to $\boldsymbol{\Sigma}$, such that $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^* + \tilde{\boldsymbol{\Sigma}}$, where $\tilde{\boldsymbol{\Sigma}}$ is the error in the covariance due to the noise. This usually makes $\boldsymbol{\Sigma}$ nonsingular. Various approaches have been developed to estimate $\mathbf{p}$ from such noisy data.

The algorithm proposed by Guidorzi (1991) makes it possible to avoid any hypothesis about $\tilde{\boldsymbol{\Sigma}}$, provided that the system studied is stationary and two independent experiments can be performed with the same covariance error $\tilde{\boldsymbol{\Sigma}}$. Let $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ be the noisy empirical covariance matrices computed from the results of these experiments. They satisfy $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_1^* + \tilde{\boldsymbol{\Sigma}}$ and $\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_2^* + \tilde{\boldsymbol{\Sigma}}$, so the noise term $\tilde{\boldsymbol{\Sigma}}$ can be eliminated by subtraction. The vector $\mathbf{p}^*$ must satisfy

$$\mathbf{p}^* \in \ker(\boldsymbol{\Sigma}_1^* - \boldsymbol{\Sigma}_2^*) = \ker(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_2).$$

If the experiments are designed so as to ensure that dim ker $(\boldsymbol{\Sigma}_1^* - \boldsymbol{\Sigma}_2^*) = 1$, one can thus estimate $\mathbf{p}^*$ uniquely, up to the normalization, by computing ker $(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_2)$. In practice, $\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_2$ will often be of full rank, because the noise will only partially be eliminated. One may then proceed by singular-value decomposition and select the direction associated with the smallest singular value.

When the effect of the noise cannot be eliminated by subtraction, various techniques can be used to estimate $\mathbf{p}$. They differ in their hypotheses about $\tilde{\boldsymbol{\Sigma}}$, which one might prefer to those underlying the least-squares method.

The *total-least-squares* method (Willems, 1986a, 1986b, 1987; Van Huffel, 1987; De Moor, 1988) assumes that all entries of $\mathbf{x}(t)$ are independently additively corrupted by noises *with the same variance* σ^2, so

$$\tilde{\mathbf{\Sigma}} = \sigma^2 \, \mathbf{I}_{n_\mathrm{p}}.$$

This variance is then estimated by the smallest singular value of $\mathbf{\Sigma}$, the estimate $\hat{\mathbf{p}}$ of $\mathbf{p}^*$ then being obtained by normalizing the associated eigenvector.

As in total least squares, the *Frisch method* (1934) assumes that all components of $\mathbf{x}(t)$ are independently additively corrupted, but allows the noise variance to depend on the component considered, so

$$\tilde{\mathbf{\Sigma}} = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \sigma_{n_\mathrm{p}}^2 \end{bmatrix}.$$

Let $\hat{\mathbf{q}}_i$ (of dimension $n_\mathrm{p} - 1$) be the unweighted least-squares solution obtained when only the ith component of $\mathbf{x}$ is assumed to be noisy ($i = 1, \dots, n_\mathrm{p}$). The role of $y(t)$ is then taken up by $-x_i(t)$, whereas the regressor vector consists of the remaining components of $\mathbf{x}$. Transform $\hat{\mathbf{q}}_i$ into an n_p-dimensional vector by inserting an ith entry equal to one. The resulting vector is then a feasible solution for $\mathbf{p}^*$, defined up to a normalization coefficient. Choose this normalization coefficient to give a unit value to some component of the vector chosen independently of i, for example the last one. Denote by $\hat{\mathbf{p}}_i$ the $(n_\mathrm{p} - 1)$-dimensional vector of the remaining components.

If $\mathbf{\Sigma}^{-1}$ can be transformed into a matrix with positive entries by changing the sign of components of $\mathbf{x}$, *i.e.* by the transformation $\mathbf{L}\mathbf{\Sigma}^{-1}\mathbf{L}$, where the signature matrix $\mathbf{L}$ is a diagonal matrix with nonzero entries equal to ±1, the vectors $\hat{\mathbf{p}}_i$ ($i = 1, \dots, n_\mathrm{p}$) all belong to the same orthant of an $(n_\mathrm{p} - 1)$-dimensional space. The set of all feasible normalized solutions then corresponds to the simplex with these vectors as vertices. A more general case has been considered by De Moor and Vandewalle (1986a, 1986b). The technique extends to dynamical systems described by recurrence equations, and then usually yields a point estimate, contrary to the static case (Beghelli, Guidorzi and Soverini, 1990).

The errors-in-variables approach will be considered again in Section 5.4.2.1, in a bounded-error context.

4.2 Least-squares based methods

When the error is not affine in $\mathbf{p}$, the least-squares method described in Section 4.1 no longer applies. Still assuming that the cost function is quadratic in the error, various approaches can compute a parameter estimate through iterative use of recursive or nonrecursive least squares. Depending on the type of model studied, many methods are obtained, which have received various names in the literature but proceed from the same philosophy. As will be mentioned, some of these methods have rather severe limitations and cannot compete with the general methods of Section 4.3. It nevertheless seems interesting to present them, because they are classically used, simple to implement, and correspond to natural ideas, the limitations of which should be made clear.

4.2.1 Pseudolinear regression

Assume that the model output can be written as

$$y_{\mathrm{m}}(t+1, \mathbf{p}) = \mathbf{r}^{\mathrm{T}}(t, \mathbf{p})\mathbf{p}.$$

If an estimate $\hat{\mathbf{p}}(t)$ of $\mathbf{p}$ is available at time t, then $\mathbf{r}(t, \mathbf{p})$ can be approximated by a vector $\tilde{\mathbf{r}}(t)$ that does not depend on $\mathbf{p}$

$$\tilde{\mathbf{r}}(t) = \mathbf{r}^{\mathrm{T}}[t, \hat{\mathbf{p}}(t)].$$

One thus gets an LP model structure

$$y_{\mathrm{mlp}}(t+1, \mathbf{p}) = \tilde{\mathbf{r}}^{\mathrm{T}}(t)\mathbf{p},$$

and $\hat{\mathbf{p}}(t+1)$ can be computed by recursive least squares. One may *hope* that this approach converges to the estimate of $\mathbf{p}$ that would have been obtained had y_{m} not been replaced by y_{mlp}. This approach can be applied to various systems.

4.2.1.1 Extended least squares

Assume that the process studied is ARMAX, described by

$$A(q, \mathbf{p}^*)y(t+1) = B(q, \mathbf{p}^*)u(t+1) + C(q, \mathbf{p}^*)\varepsilon(t+1),$$

where

$$A(q, \mathbf{p}) = 1 + a_1q^{-1} + \ldots + a_{n_{\mathrm{a}}}q^{-n_{\mathrm{a}}},$$

$$B(q, \mathbf{p}) = b_1q^{-1} + \ldots + b_{n_{\mathrm{b}}}q^{-n_{\mathrm{b}}},$$

$$C(q, \mathbf{p}) = 1 + c_1q^{-1} + \ldots + c_{n_{\mathrm{c}}}q^{-n_{\mathrm{c}}},$$

and where the ε's are i.i.d. $\mathcal{N}(0, \sigma^2)$. This amounts to saying that the following recurrence equation is satisfied:

$$y(t+1) = -a_1^*y(t) - a_2^*y(t-1) - \ldots - a_{n_{\mathrm{a}}}^*y(t+1-n_{\mathrm{a}}) + b_1^*u(t) + \ldots$$
$$+ \mathrm{b}_{n_{\mathrm{b}}}^*u(t+1-n_{\mathrm{b}}) + c_1^*\varepsilon(t) + \ldots + c_{n_{\mathrm{c}}}^*\varepsilon(t+1-n_{\mathrm{c}}) + \varepsilon(t+1).$$

The vector of the parameters to be estimated is $\mathbf{p} = (a_1, \ldots, a_{n_{\mathrm{a}}}, b_1, \ldots, b_{n_{\mathrm{b}}}, c_1, \ldots, c_{n_{\mathrm{c}}})^{\mathrm{T}}$. If the regressor vector is defined by

$$\mathbf{r}(t, \mathbf{p}^*) = [-y(t), -y(t-1), \ldots, -y(t+1-n_{\mathrm{a}}), u(t), \ldots,$$
$$u(t+1-n_{\mathrm{b}}), \varepsilon(t), \ldots, \varepsilon(t+1-n_{\mathrm{c}})]^{\mathrm{T}},$$

the equation describing the process may be written as

$$y(t+1) = \mathbf{r}^{\mathrm{T}}(t, \mathbf{p}^*)\mathbf{p}^* + \varepsilon(t+1).$$

One should indeed know the true value $\mathbf{p}^*$ of the parameters to be able to compute the past values of ε that appear in $\mathbf{r}$ from the past values of the input and output. If $\mathbf{r}(t, \mathbf{p}^*)$ were known, it would suffice to employ recursive least squares to estimate $\mathbf{p}^*$ in the maximum-likelihood sense, since the ε's are independent, stationary and Gaussian. This suggests using the approximate model

$$y_{\mathrm{mlp}}(t+1, \mathbf{p}) = \tilde{\mathbf{r}}^{\mathrm{T}}(t)\mathbf{p},$$

where the approximation $\tilde{\mathbf{r}}(t)$ of $\mathbf{r}(t, \mathbf{p}^*)$ is obtained by replacing the past innovations by the corresponding residuals

$$\tilde{\mathbf{r}}(t) = [-y(t), -y(t-1), \ldots, -y(t+1-n_{\mathrm{a}}), u(t), \ldots, \\ u(t+1-n_{\mathrm{b}}), e(t), \ldots, e(t+1-n_{\mathrm{c}})]^{\mathrm{T}},$$

where $e(t) = y(t) - y_{\mathrm{mlp}}[t, \hat{\mathbf{p}}(t)]$. All quantities appearing in $\tilde{\mathbf{r}}(t)$ can be computed at time t, so recursive least squares apply. The resulting estimator $\hat{\mathbf{p}}$ is known as the *extended-least-squares estimator* (Young, 1968; Panuska, 1968).

4.2.1.2 Properties of extended least squares

Few pseudolinear algorithms have had their convergence properties studied as exhaustively as extended least squares.

PELS1: Convergence to $\mathbf{p}^*$ is not guaranteed. A *sufficient* condition is that $\mathrm{Re}[1/C(e^{j\omega}, \mathbf{p}^*)] \geq 1/2$ for all real ω (Ljung, 1987). Since this condition depends on the true value $\mathbf{p}^*$ of the parameters, obviously unknown, it cannot be checked *a priori*. It is even possible to produce examples of ARMAX systems such that the extended-least-squares estimator will *never* converge to $\mathbf{p}^*$ (Ljung and Söderström, 1983).

PELS2: When it occurs, convergence is often rather slow anyway. Some forgetting should be introduced in the recursive algorithm, in order not to penalize the final estimate with the initial errors. If the data are too scarce, they could be circulated several times in the algorithm.

PELS3: The method can also be implemented off-line.

4.2.2 Multilinear regression

Assume that the parameter vector can be partitioned into classes $\mathbf{p}_1$, $\mathbf{p}_2$... such that the error is affine with respect to the parameters of any of these classes when the parameters of all others are fixed. It is then possible to search for $\hat{\mathbf{p}}$ by applying the least-squares method to estimate the parameters of each class in turn

$$\hat{\mathbf{p}}_1 = \arg\min j(\mathbf{p}_1 \mid \hat{\mathbf{p}}_2, \hat{\mathbf{p}}_3, \ldots),$$

$$\hat{\mathbf{p}}_2 = \arg\min j(\mathbf{p}_2 \mid \hat{\mathbf{p}}_1, \hat{\mathbf{p}}_3, \ldots),$$

and so forth, with a cyclic exploration of all classes. At each step, the value of $j(\hat{\mathbf{p}})$ decreases towards some constant value. Nothing guarantees, however, as will be seen for generalized least squares, that the estimate of $\mathbf{p}$ thus obtained corresponds to a global (or even local) minimizer of the cost.

4.2.2.1 Generalized least squares

Consider an ARARX process described by

$$A(q, \mathbf{p}^*)y(t+1) = B(q, \mathbf{p}^*)u(t+1) + \frac{1}{D(q, \mathbf{p}^*)}\,\varepsilon(t+1),$$

where A, B and ε are defined as previously and

$$D(q, \mathbf{p}) = 1 + d_1q^{-1} + \ldots + d_{n_\mathrm{d}}q^{-n_\mathrm{d}}.$$

This condensed notation corresponds to the recurrence equations

$$\begin{aligned} y(t+1) = -a_1^*y(t) - a_2^*y(t-1) - \ldots - a_{n_\mathrm{a}}^*y(t+1-n_\mathrm{a}) \\ + b_1^*u(t) + \ldots + b_{n_\mathrm{b}}^*u(t+1-n_\mathrm{b}) + \eta(t+1), \end{aligned}$$

$$\eta(t+1) = -d_1^*\eta(t) - \ldots - d_{n_\mathrm{d}}^*\eta(t+1-n_\mathrm{d}) + \varepsilon(t+1).$$

Partition the vector $\mathbf{p}$ of the parameters to be estimated into

$$\mathbf{p}_\mathrm{ab} = (a_1, \ldots, a_{n_\mathrm{a}}, b_1, \ldots, b_{n_\mathrm{b}})^\mathrm{T} \quad \text{and} \quad \mathbf{p}_\mathrm{d} = (d_1, \ldots, d_{n_\mathrm{d}})^\mathrm{T}.$$

The equation of the process becomes

$$A(q, \overset{*}{\mathbf{p}}_\mathrm{ab})y(t+1) = B(q, \overset{*}{\mathbf{p}}_\mathrm{ab})u(t+1) + \frac{1}{D(q, \overset{*}{\mathbf{p}}_\mathrm{d})}\,\varepsilon(t+1).$$

To estimate $\mathbf{p}$ in the (conditional) maximum-likelihood sense, one may use the results of Section 3.3.2. The prediction error satisfies $e_\mathrm{p}(t+1, \mathbf{p}^*) = \varepsilon(t+1)$, and is thus given by

$$e_\mathrm{p}(t+1, \mathbf{p}) = D(q, \mathbf{p}_\mathrm{d})[A(q, \mathbf{p}_\mathrm{ab})y(t+1) - B(q, \mathbf{p}_\mathrm{ab})u(t+1)].$$

Since, by hypothesis, the ε's are i.i.d. $\mathcal{N}(0, \sigma^2)$, the (conditional) maximum-likelihood estimate minimizes

$$j(\mathbf{p}) = \sum_{t=1}^{n_\mathrm{t}} e_\mathrm{p}^2(t, \mathbf{p}).$$

The prediction error e_{p} is not affine in $\mathbf{p}$, so the least-squares method cannot be used to find $\hat{\mathbf{p}}$. When $D(q, \mathbf{p}_{\mathrm{d}})$ is fixed, however, it is affine in $\mathbf{p}_{\mathrm{ab}}$. Similarly, when $A(q, \mathbf{p}_{\mathrm{ab}})$ and $B(q, \mathbf{p}_{\mathrm{ab}})$ are fixed, it is affine in $\mathbf{p}_{\mathrm{d}}$. This suggests the following algorithm:

Step 1: Set $k = 0$, $\hat{\mathbf{p}}_{\mathrm{d}}^{0} = 0$, *i.e.* $D(q, \hat{\mathbf{p}}_{\mathrm{d}}^{0}) = 1$.
Step 2: Minimize

$$j(\mathbf{p}_{\mathrm{ab}}) = \sum_{t=1}^{n_{\mathrm{t}}} [A(q, \mathbf{p}_{\mathrm{ab}})y_{\mathrm{f}}(t, \hat{\mathbf{p}}_{\mathrm{d}}^{k}) - B(q, \mathbf{p}_{\mathrm{ab}})u_{\mathrm{f}}(t, \hat{\mathbf{p}}_{\mathrm{d}}^{k})]^2$$

with respect to $\mathbf{p}_{\mathrm{ab}}$, where the filtered input u_{f} and output y_{f} are obtained by passing u and y through the filter $D(q, \hat{\mathbf{p}}_{\mathrm{d}}^{k})$, which corresponds to

$$u_{\mathrm{f}}(t, \hat{\mathbf{p}}^{k}) = u(t) + \hat{d}_{1}^{k}u(t-1) + \ldots + \hat{d}_{n_{\mathrm{d}}}^{k}u(t-n_{\mathrm{d}}),$$

$$y_{\mathrm{f}}(t, \hat{\mathbf{p}}^{k}) = y(t) + \hat{d}_{1}^{k}y(t-1) + \ldots + \hat{d}_{n_{\mathrm{d}}}^{k}y(t-n_{\mathrm{d}}).$$

The cost j is now a sum of squares of affine terms in $\mathbf{p}_{\mathrm{ab}}$. Nonrecursive least squares can therefore be used to compute the global minimizer $\hat{\mathbf{p}}_{\mathrm{ab}}^{k}$.
Step 3: Minimize

$$j(\mathbf{p}_{\mathrm{d}}) = \sum_{t=1}^{n_{\mathrm{t}}} \{D(q, \mathbf{p}_{\mathrm{d}})[A(q, \hat{\mathbf{p}}_{\mathrm{ab}}^{k})y(t) - B(q, \hat{\mathbf{p}}_{\mathrm{ab}}^{k})u(t)]\}^2$$

with respect to $\mathbf{p}_{\mathrm{d}}$. Again, this cost must be evaluated by using the recurrence equations associated with this condensed notation. It consists of a sum of squares of terms affine in $\mathbf{p}_{\mathrm{d}}$, so the global minimizer $\hat{\mathbf{p}}_{\mathrm{d}}$ can be computed by nonrecursive least squares.
Step 4: Increment k by one, set $\hat{\mathbf{p}}_{\mathrm{d}}^{k}$ equal to $\hat{\mathbf{p}}_{\mathrm{d}}$ and go to Step 2.

This method has been developed by Clarke (1967), under the name of *generalized least squares*. It alternates estimating the parameters of the deterministic part of the model (Step 2) and those of the autoregressive model of the noise (Step 3).

4.2.2.2 Properties of generalized least squares

PGLS1: Since each step decreases a quantity $j(\hat{\mathbf{p}})$ bounded from below by zero, convergence of the cost to a constant value is ensured.

PGLS2: Nothing guarantees, however, that $\hat{\mathbf{p}}$ will converge to a global (or even local) optimizer of the cost. Alternating optimizations with respect to $\mathbf{p}_{\mathrm{ab}}$ and $\mathbf{p}_{\mathrm{d}}$ actually result in a deadlock, or exceedingly slow convergence, if the minimization of j requires simultaneous action on these two classes of parameters. This is a general difficulty with all algorithms that freeze some parameters while others are optimized. See also Section 4.3.2.6 and Figure 4.12.

PGLS3: A recursive version of generalized least squares is available, with exponential forgetting of past data (Hasting-James and Sage, 1969).

4.2.3 Filtering

As for generalized least squares, many algorithms alternate filtering and use of the least-squares method.

4.2.3.1 Steiglitz and McBride's method

Consider a process described by

$$y(t+1) = y_{\mathrm{m}}(t+1, \mathbf{p}^*) + \varepsilon(t+1),$$

where the ε's are i.i.d. $\mathcal{N}(0, \sigma^2)$ and

$$y_{\mathrm{m}}(t+1, \mathbf{p}) = \frac{B(q, \mathbf{p})}{A(q, \mathbf{p})}\, u(t+1).$$

with the polynomials $A(q, \mathbf{p})$ and $B(q, \mathbf{p})$ defined as previously. This last equation is just a condensed notation for the recurrence equation

$$\begin{aligned} y_{\mathrm{m}}(t+1, \mathbf{p}) = {} & -a_1 y_{\mathrm{m}}(t, \mathbf{p}) - a_2 y_{\mathrm{m}}(t-1, \mathbf{p}) - \dots - a_{n_{\mathrm{a}}} y_{\mathrm{m}}(t+1-n_{\mathrm{a}}, \mathbf{p}) \\ & + b_1 u(t) + \dots + b_{n_{\mathrm{b}}} u(t+1-n_{\mathrm{b}}), \end{aligned}$$

where

$$\mathbf{p} = (a_1, \dots, a_{n_{\mathrm{a}}}, b_1, \dots, b_{n_{\mathrm{b}}})^{\mathrm{T}}.$$

The maximum-likelihood estimate is obtained (Chapter 3) by minimizing the cost

$$j(\mathbf{p}) = \sum_{t=1}^{n_t} [y(t) - y_{\mathrm{m}}(t, \mathbf{p})]^2,$$

which is quadratic in the *output* error

$$e_{\mathrm{y}}(t, \mathbf{p}) = y(t) - y_{\mathrm{m}}(t, \mathbf{p}).$$

Unfortunately, $y_{\mathrm{m}}(t, \mathbf{p})$ is not LP, and to use the least-squares method, one is led (Example 4.3) to change the model structure to

$$\begin{aligned} y_{\mathrm{mlp}}(t+1, \mathbf{p}) = {} & -a_1 y(t) - a_2 y(t-1) - \dots - a_{n_{\mathrm{a}}} y(t+1-n_{\mathrm{a}}) \\ & + b_1 u(t) + \dots + b_{n_{\mathrm{b}}} u(t+1-n_{\mathrm{b}}), \end{aligned}$$

in order to minimize

$$j(\mathbf{p}) = \sum_{t=1}^{n_t} [y(t) - y_{\mathrm{mlp}}(t, \mathbf{p})]^2.$$

This amounts to replacing the initial output error by a *generalized error*

$$e_g(t, \mathbf{p}) = y(t) - y_{\text{mlp}}(t, \mathbf{p}).$$

Because of this substitution, the resulting estimator $\hat{\mathbf{p}}_{\text{ls}}$ is not a maximum-likelihood estimator of $\mathbf{p}^*$. *Steiglitz and McBride's method* (1965) is intended to transform the generalized error $e_g(t, \mathbf{p})$ iteratively into the output error $e_y(t, \mathbf{p})$. The generalized error satisfies

$$e_g(t+1, \mathbf{p}) = y(t+1) - y_{\text{mlp}}(t+1, \mathbf{p}) = A(q, \mathbf{p})y(t+1) - B(q, \mathbf{p})u(t+1).$$

The difference between generalized and output errors is illustrated by Figure 4.6.

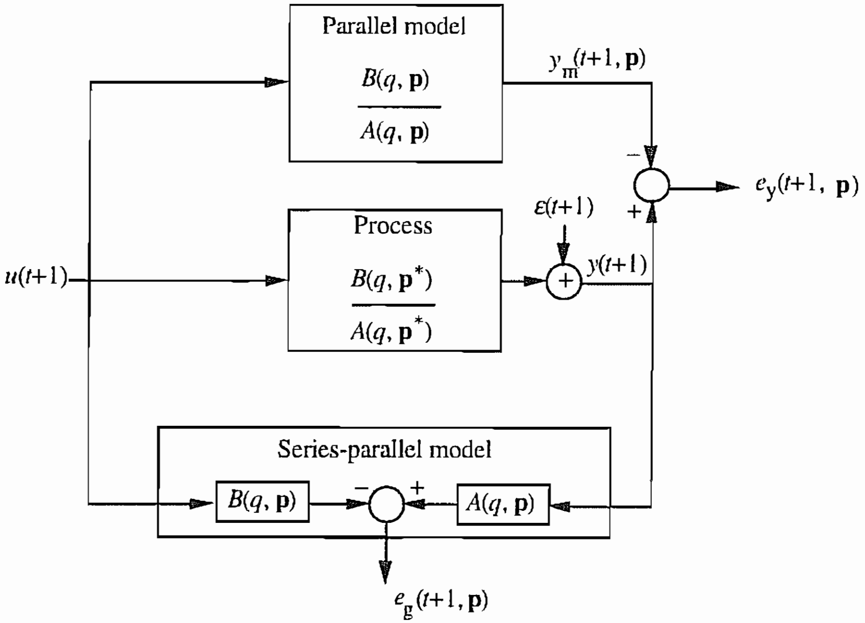


Figure 4.6. Generalized and output errors

Whereas the model producing the output error is parallel and non-LP, the one producing the generalized error is LP, with a part in series and a part in parallel. To replace the generalized error by an output error, one should filter the two inputs of the series-parallel model by $1/A(q, \mathbf{p}^*)$. Since $\mathbf{p}^*$ is unknown, the procedure is iterative (Figure 4.7). Let $\hat{\mathbf{p}}_s^k$ be the estimate of $\mathbf{p}^*$ at iteration k. The next estimate $\hat{\mathbf{p}}_s^{k+1}$ is computed by nonrecursive least squares using input and output data filtered by $1/A(q, \hat{\mathbf{p}}_s^k)$. This means that $u(t)$ and $y(t)$ are respectively replaced by $u_f(t)$ and $y_f(t)$, with

$$y_f(t+1) = -\hat{a}_1 y_f(t) - \ldots - \hat{a}_{n_a} y_f(t+1-n_a) + y(t+1),$$

$$u_f(t+1) = -\hat{a}_1 u_f(t) - \dots - \hat{a}_{n_a} u_f(t+1-n_a) + u(t+1),$$

where the $\hat{a}_i$'s are the first n_a components of $\hat{\mathbf{p}}_s^k$. The procedure is initialized by estimating $\hat{\mathbf{p}}_s^0$ by least squares from the original unfiltered data. Provided that the estimated parameters converge to a constant value,

$$\frac{A(q, \hat{\mathbf{p}}_s^{k+1})}{A(q, \hat{\mathbf{p}}_s^k)} \to 1 \text{ and } \frac{B(q, \hat{\mathbf{p}}_s^{k+1})}{A(q, \hat{\mathbf{p}}_s^k)} \to \frac{B(q, \hat{\mathbf{p}}_s^{k+1})}{A(q, \hat{\mathbf{p}}_s^{k+1})},$$

so the filtered generalized error does tend to the output error.

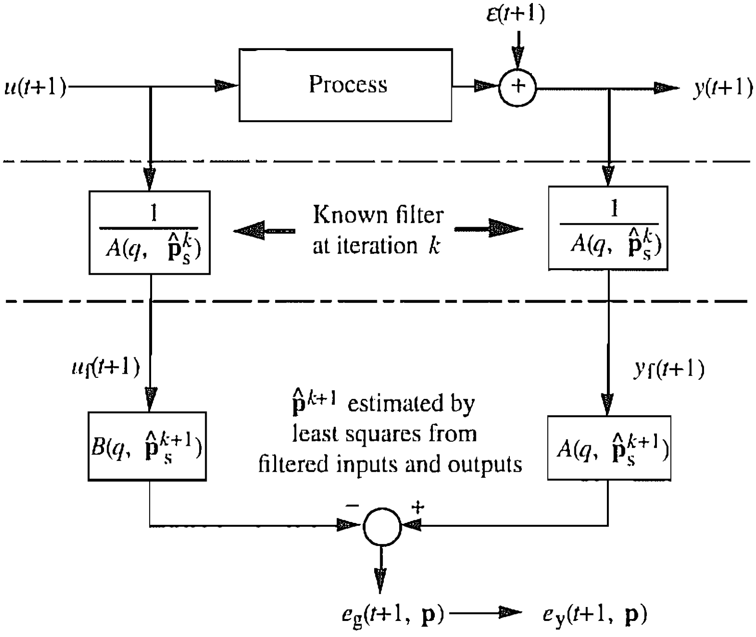

$\varepsilon(t+1)$
$u(t+1)$
Process
+
$y(t+1)$
$\frac{1}{A(q, \hat{\mathbf{p}}_s^k)}$
Known filter at iteration k
$\frac{1}{A(q, \hat{\mathbf{p}}_s^k)}$
$u_f(t+1)$
$y_f(t+1)$
$\hat{\mathbf{p}}^{k+1}$ estimated by least squares from filtered inputs and outputs
$B(q, \hat{\mathbf{p}}_s^{k+1})$
$A(q, \hat{\mathbf{p}}_s^{k+1})$
−
+
$e_g(t+1, \mathbf{p})$
$e_y(t+1, \mathbf{p})$

Figure 4.7. Principle of Steiglitz and McBride's method

Properties of Steiglitz and McBride's method

PSM1: The model structure is ARMAX, with the constraint $C(q, \mathbf{p}) = A(q, \mathbf{p})$.

PSM2: The system studied must be stable. One must also make sure that $1/A(q, \hat{\mathbf{p}}_s^k)$ is a stable filter. Otherwise, any unstable pole must be dragged back inside the unit circle to avoid explosion.

PSM3: Provided that

— $1/A(q, \mathbf{p}^*)$ is stable,
— $A(q, \mathbf{p}^*)$ and $B(q, \mathbf{p}^*)$ are relatively prime,
— the input is persistently exciting,

— the input and noise are independent, and the addditive output noise corresponds to a sequence of independent random variables,

the method converges *locally* to the true value $\mathbf{p}^*$ of the parameters as the number of data points tends to infinity. The convergence becomes global if the signal-to-noise ratio is large enough (Stoica and Söderström, 1981).

PSM4: The method usually converges very quickly (typically 3 to 4 iterations). The volume of computation is much smaller than with the nonlinear programming methods to be presented in Section 4.3. A very large number of data is not mandatory.

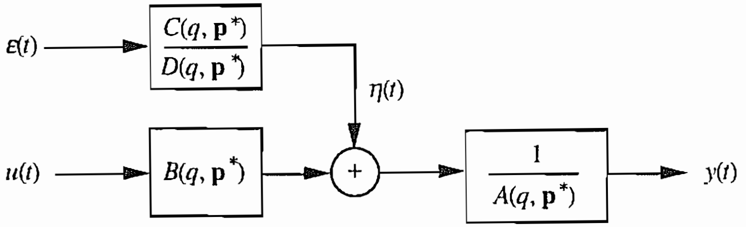
PSM5: This approach can also be implemented recursively (Ljung and Söderström, 1983).

REMARK 4.9

At each iteration, one should filter the initial input and output data (not the previously filtered data) to get u_f and y_f. ◊

4.2.3.2 Extended matrix method

All systems considered so far in Section 4.2 are special cases of ARARMAX (Figure 4.8).



$\varepsilon(t)$
$\dfrac{C(q, \mathbf{p}^*)}{D(q, \mathbf{p}^*)}$
$\eta(t)$
$u(t)$
$B(q, \mathbf{p}^*)$
+
$\dfrac{1}{A(q, \mathbf{p}^*)}$
$y(t)$

Figure 4.8. ARARMAX structure that contains all examples of Section 4.2 as special cases

Assume, as previously, that the ε's are i.i.d. $\mathcal{N}(0, \sigma^2)$. If the input-output delay n_r is assumed to be one, this scheme corresponds to

$$y(t+1) = -a_1^* y(t) - \ldots - a_{n_a}^* y(t+1-n_a) + b_1^* u(t) + \ldots + b_{n_b}^* u(t+1-n_b) + \eta(t+1),$$

$$\eta(t+1) = -d_1^* \eta(t) - \ldots - d_{n_d}^* \eta(t+1-n_d) + \varepsilon(t+1) + c_1^* \varepsilon(t) + \ldots + c_{n_c}^* \varepsilon(t+1-n_c).$$

If $\{\eta(t)\}$ and $\{\varepsilon(t)\}$ were known, the maximum-likelihood estimate of

$$\mathbf{p} = (a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b}, \ldots, c_1, \ldots, c_{n_c}, d_1, \ldots, d_{n_d})^{\mathrm{T}}$$

could be obtained by minimizing

$$j(\mathbf{p}) = \sum_{t=1}^{n_t} e_g^2(t, \mathbf{p}),$$

where

$$e_{\mathrm{g}}(t+1,\, \mathbf{p}) = y(t+1) - \mathbf{r}^{\mathrm{T}}(t)\mathbf{p},$$

with

$$\mathbf{r}(t) = [-y(t),\, \dots ,\, -y(t+1-n_{\mathrm{a}}),\, u(t),\, \dots ,\, u(t+1-n_{\mathrm{b}}),\, \varepsilon(t),\, \dots ,\\ \varepsilon(t+1-n_{\mathrm{c}}),\, -\eta(t),\, \dots ,\, -\eta(t+1-n_{\mathrm{d}})]^{\mathrm{T}}.$$

One would indeed have $e_{\mathrm{g}}(t+1, \mathbf{p}^*) = \varepsilon(t+1)$. Unfortunately, $\mathbf{r}(t)$ would only be known if $\mathbf{p}^*$ were. As in extended least squares, estimates are substituted for unknown quantities in $\mathbf{r}(t)$. From Figure 4.8, these estimates can be taken as

$$\hat{\eta}(t+1) = A(q, \hat{\mathbf{p}})y(t+1) - B(q, \hat{\mathbf{p}})u(t+1),$$

and

$$\hat{\varepsilon}(t+1) = \frac{D(q, \hat{\mathbf{p}})}{C(q, \hat{\mathbf{p}})}\hat{\eta}(t+1),$$

where $\hat{\mathbf{p}}$ is the best available estimate. One can then apply recursive least squares with forgetting or nonrecursive least squares to estimate $\mathbf{p}^*$ iteratively. This technique has been developed by Talmon and van den Boom (1973) and is known under the (admittedly not too explicit) name of the *extended matrix method* (Eykhoff, 1974).

Properties of the extended matrix method

PEM1: To avoid explosion, all zeros of $C(q, \hat{\mathbf{p}})$ must be forced to stay inside the unit circle.

PEM2: Characterization becomes very complicated, since n_{a}, n_{b}, n_{c} and n_{d} must be chosen.

PEM3: A large number of data points is necessary.

PEM4: The convergence of the method is not guaranteed (Ljung, Söderström and Gustavsson, 1975), and a more general method with better convergence properties will be presented in Section 4.3.8.

PEM5: In the special case of ARMAX structures ($D(q, \mathbf{p}) = 1$), this method differs from extended least squares by a more sophisticated substitution for ε in the regressor vector.

4.2.4 First-order expansion of the error

Since, by hypothesis,

$$j(\mathbf{p}) = \frac{1}{2}\,\mathbf{e}^{\mathrm{T}}(\mathbf{p})\mathbf{Q}\mathbf{e}(\mathbf{p}),$$

another method of transforming the problem into one that can be solved by least squares is to perform a first-order expansion of the error $\mathbf{e}$ around the last estimate $\hat{\mathbf{p}}^k$:

$$\mathbf{e}(\hat{\mathbf{p}}^{k+1}) \simeq \mathbf{e}(\hat{\mathbf{p}}^{k}) + \frac{\partial \mathbf{e}}{\partial \mathbf{p}^{\mathrm{T}}}\Big|_{\hat{\mathbf{p}}^k}(\hat{\mathbf{p}}^{k+1} - \hat{\mathbf{p}}^{k}).$$

This approximation of the error is affine in $\hat{\mathbf{p}}^{k+1} - \hat{\mathbf{p}}^k$. The least-squares method can therefore be used to compute the increment to be given to $\hat{\mathbf{p}}$ at iteration k, which yields

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \Big[\frac{\partial \mathbf{e}^{\mathrm{T}}(\mathbf{p})}{\partial \mathbf{p}}\Big|_{\hat{\mathbf{p}}^k} \mathbf{Q} \frac{\partial \mathbf{e}(\mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}\Big|_{\hat{\mathbf{p}}^k}\Big]^{-1} \frac{\partial \mathbf{e}^{\mathrm{T}}(\mathbf{p})}{\partial \mathbf{p}}\Big|_{\hat{\mathbf{p}}^k} \mathbf{Q}\mathbf{e}(\hat{\mathbf{p}}^k).$$

This is the *Gauss-Newton algorithm*. When $\mathbf{Q}$ is diagonal, with nonzero entries w_t $(t = 1, \ldots, n_t)$, it becomes

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \left[\sum_{t=1}^{n_t} w_t \frac{\partial e(t, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\hat{\mathbf{p}}^k} \frac{\partial e(t, \mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}\Big|_{\hat{\mathbf{p}}^k}\right]^{-1} \sum_{t=1}^{n_t} w_t\, e(t, \hat{\mathbf{p}}^k) \frac{\partial e(t, \mathbf{p})}{\partial \mathbf{p}}\Big|_{\hat{\mathbf{p}}^k}.$$

In this form, the Gauss-Newton algorithm is not even guaranteed to converge to a local minimizer of the cost. This defect will be eliminated in Section 4.3.3.4.

4.2.5 Instrumental-variable method

Let $\mathbf{y}^{\mathrm{s}}$ be a vector of data generated by a system with parameters $\mathbf{p}^*$:

$$\mathbf{y}^{\mathrm{s}} = \mathbf{R}\mathbf{p}^* + \mathbf{n}.$$

The unweighted least-squares estimator of $\mathbf{p}^*$ satisfies

$$\hat{\mathbf{p}}_{\mathrm{ls}} = (\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{y}^{\mathrm{s}} = \mathbf{p}^* + (\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{n}.$$

This estimator is unbiased, *i.e.* its mean over all data that could be collected from the system with parameters $\mathbf{p}^*$ is equal to the true value,

$$\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}\{\hat{\mathbf{p}}_{\mathrm{ls}}\} = \mathbf{p}^*,$$

if and only if

$$\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}\{(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{n}\} = \mathbf{0}.$$

This will hold true if $\mathbf{n}$ is zero-mean and uncorrelated with $\mathbf{R}$. The following example illustrates a situation where this hypothesis of uncorrelatedness is not satisfied.

EXAMPLE 4.8

Consider the data generated by the system

$$y_{\mathrm{m}}(t+1, \mathbf{p}^*) = -\overset{*}{a}_1 y_{\mathrm{m}}(t, \mathbf{p}^*) - \ldots - \overset{*}{a}_{n_{\mathrm{a}}} y_{\mathrm{m}}(t+1-n_{\mathrm{a}}, \mathbf{p}^*) + \overset{*}{b}_1 u(t) + \ldots + \overset{*}{b}_{n_{\mathrm{b}}} u(t+1-n_{\mathrm{b}}),$$
$$y(t) = y_{\mathrm{m}}(t, \mathbf{p}^*) + \varepsilon(t),$$

where $\varepsilon(t)$ belongs to a sequence of independent random variables and

$$\mathbf{p} = (a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b})^T.$$

The data satisfy

$$y(t+1) = \mathbf{r}^T(t)\mathbf{p}^* + n(t+1),$$

with

$$\mathbf{r}(t) = [-y(t), \ldots, -y(t+1-n_a), u(t), \ldots, u(t+1-n_b)]^T$$

and

$$n(t+1) = \varepsilon(t+1) + a_1^* \varepsilon(t) + \ldots + a_{n_a}^* \varepsilon(t+1-n_a).$$

However, since the regressor vector can also be written

$$\mathbf{r}(t) = [-y_m(t, \mathbf{p}^*) - \varepsilon(t), \ldots, -y_m(t+1-n_a, \mathbf{p}^*) - \varepsilon(t+1-n_a), u(t), \ldots, u(t+1-n_b)]^T,$$

R is correlated with **n**, so the least-squares estimator is biased. ◊

Note that the unweighted least-squares estimator is obtained by left-multiplying

$$\mathbf{y}^s = \mathbf{R}\hat{\mathbf{p}}_{ls} + \hat{\mathbf{n}}_{ls}$$

by $\mathbf{R}^T$ then imposing $\mathbf{R}^T\hat{\mathbf{n}}_{ls} = \mathbf{0}$. The basic idea of the instrumental-variable method is to obtain $\hat{\mathbf{p}}_{iv}$ by left-multiplying

$$\mathbf{y}^s = \mathbf{R}\hat{\mathbf{p}}_{iv} + \hat{\mathbf{n}}_{iv}$$

by $\mathbf{V}^T$ then imposing $\mathbf{V}^T\hat{\mathbf{n}}_{iv} = \mathbf{0}$. Assuming that $\mathbf{V}^T\mathbf{R}$ is invertible, one then gets

$$\hat{\mathbf{p}}_{iv} = (\mathbf{V}^T\mathbf{R})^{-1}\mathbf{V}^T\mathbf{y}^s.$$

The entries of **V** are the *instrumental variables*, or *instruments*. The estimator $\hat{\mathbf{p}}_{iv}$ satisfies

$$\hat{\mathbf{p}}_{iv} = \mathbf{p}^* + (\mathbf{V}^T\mathbf{R})^{-1}\mathbf{V}^T\mathbf{n}.$$

It will be unbiased if **n** is zero-mean and uncorrelated with **V**.

The quality of the results depends on that of the instruments, and various methods have been proposed to generate them (Söderström and Stoica, 1983). The simplest solution is to obtain **V** by replacing all entries in **R** that may be correlated with **n** by the output of a run of the model at the previous iteration. One thus gets an iterative procedure

$$\hat{\mathbf{p}}_{iv}^{k+1} = [\mathbf{V}^T(\hat{\mathbf{p}}_{iv}^k)\ \mathbf{R}]^{-1}\mathbf{V}^T(\hat{\mathbf{p}}_{iv}^k)\mathbf{y}^s,$$

which may be initialized by least squares.

EXAMPLE 4.8 (continued)

To obtain **V** from **R**, one may replace each row $\mathbf{r}^T(t)$ of **R** by the instruments

$$\mathbf{v}^T(t, \hat{\mathbf{p}}_{iv}^k) = [-y_m(t, \hat{\mathbf{p}}_{iv}^k), \ldots, -y_m(t+1-n_a, \hat{\mathbf{p}}_{iv}^k), u(t), \ldots, u(t+1-\mathrm{n_b})],$$

where y_{m} is obtained by simulating the recurrence equation

$$y_{\mathrm{m}}(t+1, \mathbf{p}) = -a_1 y_{\mathrm{m}}(t, \mathbf{p}) - \ldots - a_{n_{\mathrm{a}}} y_{\mathrm{m}}(t+1-n_{\mathrm{a}}, \mathbf{p}) + b_1 u(t) + \ldots + b_{n_{\mathrm{b}}} u(t+1-n_{\mathrm{b}}). \quad \lozenge$$

The instrumental-variable method can also be implemented recursively, similarly to least squares. Its main interest is that it allows the bias of the least-squares method to be avoided, without being too specific about the nature of the process generating the noise (contrary to generalized or extended least squares, for example).

REMARK 4.10

Being unbiased is not necessarily a good property of an estimator. It is sometimes better to accept some bias, if this leads to an important reduction of the variance of the estimation error. This is the aim of the *ridge estimator*,

$$\hat{\mathbf{p}} = (\mathbf{R}^{\mathrm{T}}\mathbf{R} + \mathbf{K})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{y}^{\mathrm{s}},$$

where the matrix $\mathbf{K}$ is symmetrical and chosen so as to reduce the estimation mean-square error; see, *e.g.*, (Goldstein and Smith, 1974). $\lozenge$

4.2.6 Least squares on correlations

Consider a process described by

$$y(t+1) = -a_1^* y(t) - \ldots - a_{n_{\mathrm{a}}}^* y(t+1-n_{\mathrm{a}}) + b_1^* u(t) + \ldots + b_{n_{\mathrm{b}}}^* u(t+1-n_{\mathrm{b}}) + \eta(t+1),$$

where the noise η is assumed to be uncorrelated with the input u. The input autocorrelation and input-output crosscorrelation, respectively defined by

$$c_{\mathrm{uu}}(t) = \lim_{n\to\infty} \frac{1}{n+1} \sum_{k=0}^{n} u(k)u(k-t) \quad \text{and} \quad c_{\mathrm{yu}}(t) = \lim_{n\to\infty} \frac{1}{n+1} \sum_{k=0}^{n} y(k)u(k-t),$$

satisfy

$$\begin{aligned} c_{\mathrm{yu}}(t+1) = &-a_1^* c_{\mathrm{yu}}(t) - \ldots - a_{n_{\mathrm{a}}}^* c_{\mathrm{yu}}(t+1-n_{\mathrm{a}}) \\ &+ b_1^* c_{\mathrm{uu}}(t) + \ldots + b_{n_{\mathrm{b}}}^* c_{\mathrm{uu}}(t+1-n_{\mathrm{b}}) + c_{\eta\mathrm{u}}(t+1), \end{aligned}$$

where the crosscorrelation $c_{\eta\mathrm{u}}$ between u and η is zero. The influence of the noise can therefore be eliminated by applying least squares to the LP model structure

$$c_{\mathrm{yu_m}}(t+1, \mathbf{p}) = [-c_{\mathrm{yu}}(t), \ldots, -c_{\mathrm{yu}}(t+1-n_{\mathrm{a}}), c_{\mathrm{uu}}(t), \ldots, c_{\mathrm{uu}}(t+1-n_{\mathrm{b}})]\mathbf{p}$$

with the quadratic cost

$$j(\mathbf{p}) = \sum_t [c_{\mathrm{yu}}(t) - c_{\mathrm{yu_m}}(t, \mathbf{p})]^2.$$

Large values of n are required to get accurate empirical correlations from experimental data (provided that the process generating the data is sufficiently stationary). This makes this technique rather inefficient. It can be implemented recursively or nonrecursively (Isermann, 1974).

4.3 General methods

Many optimization methods do not require the cost to be quadratic or the error to be affine in the parameters; see, *e.g.*, Polak (1971), Luenberger (1973), Powell (1981), Gill, Murray and Wright (1981), Minoux (1983), Press *et al.* (1986), Polyak (1987) and Lemaréchal (1989). We shall only present a few of them. The techniques selected fall into three categories. The first corresponds to basic tools, such as the gradient method or one-dimensional search methods, that are the core of more sophisticated algorithms. The second corresponds to reference algorithms that have a proven record on a number of applications. Some of them, such as conjugate-gradient or quasi-Newton methods, are implemented in most major libraries of scientific subroutines. Others, such as Levenberg and Marquardt's method, are used in commercially available estimation software. The third group of algorithms consists of methods still under development but very promising, such as global optimization methods.

The initial problem is generally decomposed into a sequence of more elementary problems. For instance, a multivariable optimization problem is considered as a sequence of one-dimensional optimizations, or optimization on a convex set is treated as a sequence of optimizations under linear constraints. Most of these subproblems cannot be solved exactly, in the sense that the algorithms include some stopping rules such as "if $f(\mathbf{p})$ is lower than some threshold, stop". The problem of how to choose thresholds in imbricated algorithms so as to ensure convergence of the whole procedure will not be considered here. For a rigorous exposition, see (Polak, 1971). Similarly, the numerical implementation of the algorithms will not be described, and the reader is invited to consult the many references mentioned.

Except for Section 4.3.4, all techniques to be considered deal with basically unconstrained optimization, even if the prior feasible domain $\mathbb{P}$ is sometimes assumed to be an axis-aligned orthotope (or box). To simplify exposition, the cost function j will be assumed to be minimized with respect to $\mathbf{p}$. This is not restrictive, since changing the sign of a function to be maximized makes it a cost to be minimized. A (possibly local) minimizer of j will be denoted by $\hat{\mathbf{p}}$, whereas a global minimizer will be denoted by $\check{\mathbf{p}}$. Whenever the derivative of a function with respect to its arguments is to be computed, it will be implicitly assumed to exist. Non-differentiable costs, such as those involving absolute values, will be considered in Section 4.3.5.

At each iteration k, most algorithms compute $\hat{\mathbf{p}}^{k+1}$ from $\hat{\mathbf{p}}^k$ to ensure that $j(\hat{\mathbf{p}}^{k+1})$ is lower than $j(\hat{\mathbf{p}}^k)$. This raises the questions of where to start (choice of $\hat{\mathbf{p}}^0$) and when to stop the iterative procedure. They will be addressed in Sections 4.3.6 and 4.3.7. Section 4.3.8 will deal with recursive techniques, by which data can be taken into account as they arrive. Section 4.3.9 will be devoted to global optimization.

Before presenting these methods, let us consider a common special case where the least-squares method can still be used to simplify the optimization problem considerably.

4.3.1 Quadratic cost and partially LP structure

Consider a model structure, the output of which depends linearly on n_l parameters forming a vector $\mathbf{p}^l$ and nonlinearly on n_{nl} parameters forming a vector $\mathbf{p}^{nl}$. The vector of all model outputs can be written as

$$\mathbf{y}^m(\mathbf{p}^l, \mathbf{p}^{nl}) = \mathbf{R}(\mathbf{p}^{nl})\mathbf{p}^l.$$

Assume that the quadratic cost

$$j(\mathbf{p}^l, \mathbf{p}^{nl}) = [\mathbf{y}^s - \mathbf{y}^m(\mathbf{p}^l, \mathbf{p}^{nl})]^T\mathbf{Q}[\mathbf{y}^s - \mathbf{y}^m(\mathbf{p}^l, \mathbf{p}^{nl})],$$

where $\mathbf{Q} \geq 0$, is to be minimized. For any fixed value of $\mathbf{p}^{nl}$, the value of $\mathbf{p}^l$ that minimizes $j(., \mathbf{p}^{nl})$ is given by the least-squares method as a function of $\mathbf{p}^{nl}$:

$$\hat{\mathbf{p}}^l(\mathbf{p}^{nl}) = [\mathbf{R}^T(\mathbf{p}^{nl})\mathbf{Q}\mathbf{R}(\mathbf{p}^{nl})]^{-1}\mathbf{R}^T(\mathbf{p}^{nl})\mathbf{Q}\mathbf{y}^s.$$

Replacing $\mathbf{p}^l$ by $\hat{\mathbf{p}}^l(\mathbf{p}^{nl})$ in the cost, one gets a cost that depends only on the nonlinear parameters (Lawton and Sylvestre, 1971)

$$j(\mathbf{p}^{nl}) = [\mathbf{y}^s - \mathbf{R}(\mathbf{p}^{nl})\hat{\mathbf{p}}^l(\mathbf{p}^{nl})]^T\mathbf{Q}[\mathbf{y}^s - \mathbf{R}(\mathbf{p}^{nl})\hat{\mathbf{p}}^l(\mathbf{p}^{nl})].$$

This makes it possible to reduce the dimension of the search space from $n_l + n_{nl}$ to n_{nl}, which simplifies the task considerably. We need now only provide an initial value for the nonlinear parameters. Moreover, exploring a reduced-dimension space is quicker and raises fewer numerical difficulties.

This method should be preferred to alternating minimizations with respect to $\mathbf{p}^l$ and $\mathbf{p}^{nl}$, because it avoids the deadlocks encountered in that case. For practical advice on the minimization method to compute $\hat{\mathbf{p}}^{nl}$, see (Barham and Drane, 1972; Golub and Pereyra, 1973).

EXAMPLE 4.9

Consider the model

$$y_m(t, \mathbf{p}) = \sum_{i=1}^{3} a_i \exp(-\lambda_i t),$$

where the parameters are the a_i's and λ_i's. The model output is linear in the a_i's and nonlinear in the λ_i's. If the cost to be minimized is quadratic in output error, this method makes it possible to search for the optimal value of the parameters in a three-dimensional space. ◊

4.3.2 One-dimensional optimization

Minimizing a univariate cost forms an essential component of many multidimensional search methods. Let p be the scalar parameter with respect to which j is to be minimized. It may, for instance, be associated with some direction of parameter space, spanned by some vector $\mathbf{d}$, to be explored from the estimate $\hat{\mathbf{p}}^k$ obtained at iteration k. It will thus be assumed that

$$\mathbf{p}(p) = \hat{\mathbf{p}}^k + p\mathbf{d}.$$

REMARK 4.11

For some of the multidimensional optimization methods to be presented in Section 4.3.3, it is more efficient to perform rather inaccurate one-dimensional searches, only intended to provide a significant decrease of the cost in as few iterations as possible. See Wolfe's method, to be presented in Section 4.3.3.9, which is incapable of locating the optimizer precisely and thus not considered as a one-dimensional optimization method. ◊

4.3.2.1 Definition of a search interval

The first task to be performed is to define some initial interval $\mathbb{I}^0 = [a, b]$ within which the search for $\hat{p}$ is to take place. For this purpose, one may for instance evaluate j and its first derivative with respect to the scalar p at $p^0 = 0$. (If this derivative is not available, it could be replaced by a finite difference.) Three cases may then arise, depending on the sign of this derivative.

— When the derivative is negative, the cost can be reduced by increasing p from $a = p^0$. The cost is then evaluated at $p^1 = p^0 + \Delta p$, where $\Delta p > 0$. If $j(p^1) \geq j(p^0)$, then $b = p^1$, and j possesses at least one (possibly local) minimizer between a and b. Else, one should move further away from p^0, *e.g.* by doubling Δp, until the cost starts increasing. The last and last-but-two values of p for which the cost has been evaluated can then be taken as b and a respectively. The search interval can be further reduced if the sign of the derivative of j at the last-but-one value of p is known (this value of p can then be taken as a or b).

— When the derivative is positive, the cost can be decreased by reducing p from $b = p^0$. The cost is then evaluated at $p^1 = p^0 - \Delta p$, where $\Delta p > 0$. If $j(p^1) \geq j(p^0)$, then $a = p^1$. Else, one should move further away from p^0 until the cost starts increasing. The last and last-but-two values of p for which the cost has been evaluated can then be taken as a and b respectively. Again, the search interval can be further reduced if the sign of the derivative of j at the last but one value of p is known.

— When the derivative is zero, the cost is stationary at p^0, which may be a minimizer. The cost may then be evaluated at two neighbouring points to get more information on the nature of this stationary point. If it turns out not to correspond to a minimum, it suffices to move slightly away from p^0 to get into one of the two previous cases.

REMARK 4.12

This policy leads to moving in the opposite direction to that of the gradient of the cost with respect to p. The gradient algorithm will use the same idea in a multidimensional setting. ◊

Once a search interval has been defined, various policies can be used to reduce its size. They can be compared by the volume of computation required to locate the minimizer with a given precision, which should be kept as small as possible.

4.3.2.2 Dichotomy

This method evaluates the derivative of the cost (or an approximation of this derivative by a finite difference) in the middle μ of the interval $[a, b]$. If this derivative is positive, b is replaced by μ. Else, a is replaced by μ (Figure 4.9). Each iteration therefore divides the search interval by two, so after N evaluations of the *derivative* of the cost one has

$$\text{Final uncertainty about } \hat{\mathbf{p}} = \frac{\text{initial uncertainty}}{2^N}.$$

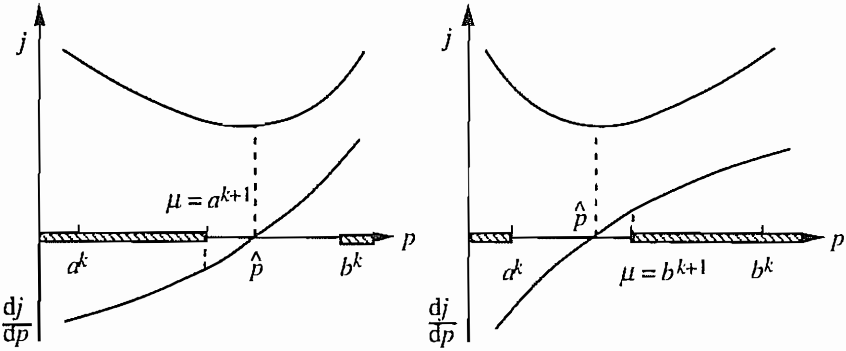


Figure 4.9. Dichotomous search

Even when the initial search interval is very large, only a few iterations are needed to locate a minimizer with high accuracy. If the function is inverse unimodal over $[a, b]$, *i.e.* if it possesses a single minimizer in $[a, b]$, this global minimizer is guaranteed to belong to the final interval. Unfortunately, j is seldom known to be inverse unimodal over $[a, b]$, so the minimizer guaranteed to be enclosed in the final interval may only be local. It may even correspond to a value of the cost larger than $j(\hat{\mathbf{p}}^k)$ (Figure 4.10).

4.3.2.3 Fibonacci's and golden-section methods

If the cost function is not differentiable, or if evaluating its derivative with respect to p is too costly or too inaccurate, dichotomy can no longer be used. The Fibonacci and golden-section methods (Kiefer, 1953, 1956) replace evaluating the derivative in the middle of the search interval $\mathbb{I}^k = [a^k, b^k]$ by comparing the values of the cost at two points p_1^k and p_2^k in this interval, with $p_2^k > p_1^k$. $\mathbb{I}^k$ is thus divided into three parts. Assuming that j is inverse unimodal over $[a, b]$, one can then use the following algorithm:

$$\text{If } j(p_2^k) > j(p_1^k), \text{ then } a^{k+1} = a^k \text{ and } b^{k+1} = p_2^k, \text{ else } a^{k+1} = p_1^k \text{ and } b^{k+1} = b^k.$$

This procedure is illustrated by Figure 4.11.

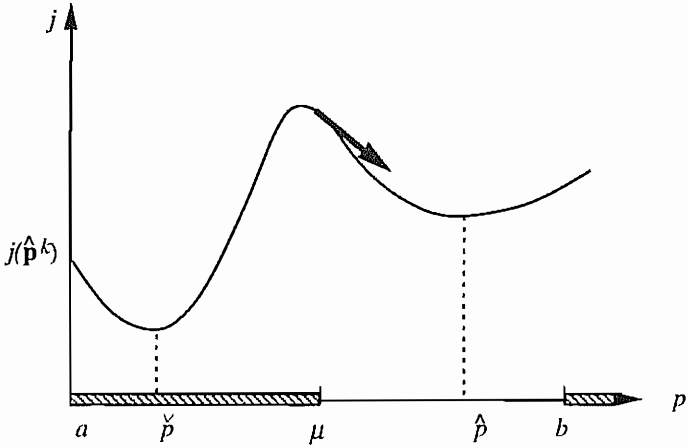


Figure 4.10. Failure of a dichotomous search due to inverse multimodality

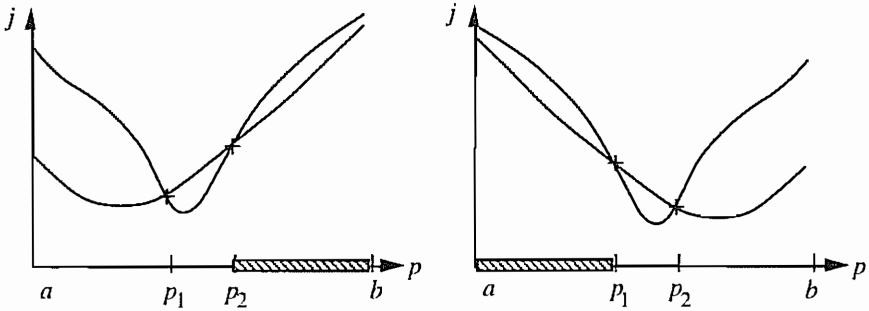


Figure 4.11. Principle of the Fibonacci and golden-section methods

The interval $\mathbb{I}^{k+1}$ therefore still contains either p_1^k or p_2^k. This suggests one additional evaluation of the cost per iteration, to be used together with the value at this remaining point. We shall limit ourselves here to a classical description of these optimization methods, which relies on minimax optimality. The final length of the search interval is minimized in the worst case, asymptotically with the golden-section method and for a given finite number N of evaluations of the cost with the Fibonacci method. More recent approaches based on average optimality seem promising; see (Wynn and Zhigljavsky, 1993) for the asymptotic case and (Pronzato and Zhigljavsky, 1993) when N is finite.

Assume first that N is finite. Let L_k be the length of the search interval $\mathbb{I}^k$ obtained after k evaluations of the cost in $\mathbb{I}^0 = [a^0, b^0]$. Since a single evaluation does not allow any part of the interval to be eliminated, $L_1 = L_0 = b^0 - a^0$. A strategy is defined by the choice of p^{k+1} as a function of $\mathbb{I}^k$ and the point p^k of $\mathbb{I}^k$ where the cost has already been evaluated. Finding the optimal strategy can be viewed as a dynamic-programming problem, with a terminal cost given by $\sup_{j \in \mathbb{J}} L_N$, where $\mathbb{J}$ is the class of all cost functions inverse unimodal over $\mathbb{I}^0$. The direct problem is to find the sequence $\{p^k\}$, with k increasing from 1 to N, but the optimal solution is obtained backwards, with k decreasing from N to 1.

Let $\mathbb{I}^{N-1} = [a^{N-1}, b^{N-1}]$. The optimal locations of the last two points can be taken as

$$p^{N-1^*} = \frac{1}{2}(a^{N-1} + b^{N-1}), \quad \text{and} \quad p^{N^*} = p^{N-1^*} + \varepsilon,$$

so that

$$L_N = \frac{1}{2} L_{N-1} + \varepsilon,$$

with ε chosen as small as possible, limited by the imprecision with which the cost is evaluated. In what follows, it will be assumed that $\varepsilon = 0$. The search interval can be normalized by transforming $\mathbb{I}^k$ to $[0, 1]$, so that p^k becomes

$$z^k = \frac{p^k - a^k}{b^k - a^k}.$$

The optimal strategy for $N = 2$ is then defined by $z^{1^*} = z^{2^*} = 1/2$. Decreasing k, one shows that the optimal strategy satisfies

$$L_{k-1}^* = L_k^* + L_{k+1}^*,$$

i.e. $L_{N-1}^* = 2L_N^*$, $L_{N-2}^* = 3L_N^*$, $L_{N-3}^* = 5L_N^*$, $L_{N-4}^* = 8L_N^* \ldots$ which is continued up to L_1^*. In the renormalized interval $[0, 1]$, the cost must then be evaluated at z^{k^*} or $1 - z^{k^*}$, with

$$z^{k^*} = \frac{L_k^* - L_{k+2}^*}{L_k^*} = \frac{L_{k+1}^*}{L_k^*}, \; k \geq 1,$$

i.e.

$$z^{N^*} = z^{N-1^*} = \frac{1}{2}, \; z^{N-2^*} = \frac{2}{3}, \; z^{N-3^*} = \frac{3}{5}, \; z^{N-4^*} = \frac{5}{8} \ldots$$

up to z^{1^*}. Note that this is a symmetrical algorithm, for in each interval $\mathbb{I}^k$ the two evaluations are performed symmetrically with respect to the centre $(a^k + b^k)/2$. The resulting method has Fibonacci's name because

$$L_{N-k}^* = f_{k+2} L_N^*,$$

with $\{f_k\}$ the sequence of Fibonacci numbers,

$$f_0 = 0, \; f_1 = 1, \; f_k = f_{k-1} + f_{k-2}, \; k \geq 2.$$

For fixed N and $\mathbb{I}^0$, this method minimizes the length L_N^* of the final interval for the worst possible cost function.

To facilitate comparison with the other methods, it is useful to get an approximate expression for the ratio $L_0/L_N^* = f_{N+1}$. The recurrence equation $f_k = f_{k-1} + f_{k-2}$ can also be written in state-space form as $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k$, with

$$\mathbf{x}_k = \begin{bmatrix} f_k \\ f_{k+1} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{x}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The eigenvalues of **A** are $1 + \alpha$ and $-\alpha$, with $\alpha = (\sqrt{5} - 1)/2$ the golden number. When k gets large enough, the contribution of the stable mode can be neglected, so

$$\frac{L_0}{L_N^*} \approx \frac{(1 + \alpha)^{N+1}}{\sqrt{5}} \approx \frac{1.6180^{N+1}}{\sqrt{5}}.$$

The Fibonacci method requires the number N of evaluations of the cost to be fixed in advance. This is not the case for the golden-section method, which corresponds to the asymptotic limit of the Fibonacci method when N tends to infinity. It can easily be checked that

$$\lim_{k \longrightarrow \infty} z^{k-1^*} = \lim_{k \longrightarrow \infty} \frac{L_k^*}{L_{k-1}^*} = \frac{1}{1 + \alpha} = \alpha\,.$$

The successive points where the cost is evaluated are therefore located within a fraction α of the extremities of the interval, *i.e.*

$$p^k = a^k + \alpha(b^k - a^k) \text{ or } p^k = a^k + (1 - \alpha)(b^k - a^k),$$

which is why the method is called the golden-section algorithm. After N evaluations of the cost, one gets

$$\frac{L_0}{L_N} = \frac{1}{\alpha^{N-1}} = (1 + \alpha)^{N-1}.$$

4.3.2.4 Parabolic interpolation

In the neighbourhood of a minimizer $\hat{p}$, the cost can often be approximated by a parabola. Given the values taken by the cost at three points p_1, p_2 and p_3 of the search interval $[a, b]$, one can compute the parabola $P(p)$ that takes exactly the same values at these three points by the Lagrange interpolation formula:

$$P(p) = j(p_1)\frac{(p - p_2)(p - p_3)}{(p_1 - p_2)(p_1 - p_3)}$$

$$+ j(p_2)\frac{(p - p_1)(p - p_3)}{(p_2 - p_1)(p_2 - p_3)} + j(p_3)\frac{(p - p_1)(p - p_2)}{(p_3 - p_1)(p_3 - p_2)},$$

then the value of p that corresponds to the stationary point of this parabola:

$$\hat{p}_{\mathrm{pi}} = p_2 - \frac{1}{2}\frac{(p_2 - p_1)^2[j(p_2) - j(p_3)] - (p_2 - p_3)^2[j(p_2) - j(p_1)]}{(p_2 - p_1)[j(p_2) - j(p_3)] - (p_2 - p_3)[j(p_2) - j(p_1)]}\,.$$

$\hat{p}_{\mathrm{pi}}$ can then be used as an approximation to the minimizer of the cost (after checking that it does not correspond to a maximum of the parabola!)

If the cost function is sufficiently uncooperative, this may lead to absurd results. When, on the other hand, a quadratic approximation is suitable, this method converges much more quickly than the previous ones.

4.3.2.5 Which method?

All these methods rely on the hypothesis that the cost $j(p)$ is inverse unimodal over $[a, b]$. If the evaluation of $\mathrm{d}j/\mathrm{d}p$ is of the same order of complexity as that of the cost, dichotomy is *much more efficient* than the Fibonacci and golden-section methods. If, on the other hand, $\mathrm{d}j/\mathrm{d}p$ is evaluated by finite difference *via* two evaluations of the cost, the Fibonacci and golden-section methods become more efficient than dichotomy.

The Fibonacci method is always more efficient than the golden-section method, unless the total number of evaluations of the cost is changed during the search.

As an illustration, Table 4.1 gives the factor by which the length of the initial search interval is divided after 10 evaluations of the cost ($N = 10$).

Dichotomy, with evaluation $\frac{\mathrm{d}j}{\mathrm{d}p} \sim$ that of j	Dichotomy, with evaluation $\frac{\mathrm{d}j}{\mathrm{d}p} \sim 2 \times$ that of j	Fibonacci	Golden section
1024 (= 2^{10})	32 (= 2^5)	89	≈ 76

Table 4.1. Comparison of performance of one-dimensional search methods

Whatever the method selected, one should not ask for more accuracy than necessary, to avoid needless evaluations of the cost.

Parabolic interpolation may accelerate search drastically, but is not sufficiently reliable to be employed alone. It should rather be used in conjunction with some type of interval elimination. This may be done with or without using the derivative of the cost with respect to p (Brent, 1973; Press *et al.*, 1986).

4.3.2.6 Combining one-dimensional optimizations

To minimize a cost j with respect to a vector $\mathbf{p}$ comprising n_p scalar parameters, one may consider a sequence of one-dimensional minimizations with respect to each of the entries of $\mathbf{p}$. The first idea that comes to mind is to explore all parameters cyclically:

$$\hat{p}_i = \arg \min_{p_i} j(p_i, \{\hat{p}_k \mid k \neq i\}),\ i = 1, \ldots, n_\mathrm{p}.$$

However, if the cost has a valley not oriented along a parameter axis, this technique will take tinier and tinier steps, resulting in a very slow exploration of the valley (Figure 4.12).

In such a case, the direction of the one-dimensional search should be modified to allow explorations along the valley. This is the purpose of *Powell's method* (1981) which, in its simplest version, proceeds as follows:

Step 1: Starting from $\hat{\mathbf{p}}^k$, find $\hat{\mathbf{p}}^{k+}$ by performing $n_\mathrm{p} = \dim \mathbf{p}$ one-dimensional minimizations in linearly independent directions $\mathbf{d}_i$ ($i = 1, \ldots, n_\mathrm{p}$).

Step 2: Find $\hat{\mathbf{p}}^{k+1}$ by performing an $(n_\mathrm{p} + 1)$th minimization in the direction

$$\mathbf{d}_{n_\mathrm{p}+1} = \hat{\mathbf{p}}^{k+} - \hat{\mathbf{p}}^k.$$

Step 3: Let $\hat{\mathbf{d}}$ be the direction $\mathbf{d}_i$ associated with the *largest* decrease of the cost in Step 1. Replace $\hat{\mathbf{d}}$ by $\mathbf{d}_{n_\mathrm{p}+1}$. Increment k by one and go to Step 1.

For the first iteration, the $\mathbf{d}_i$'s correspond to the axes of parameter space.

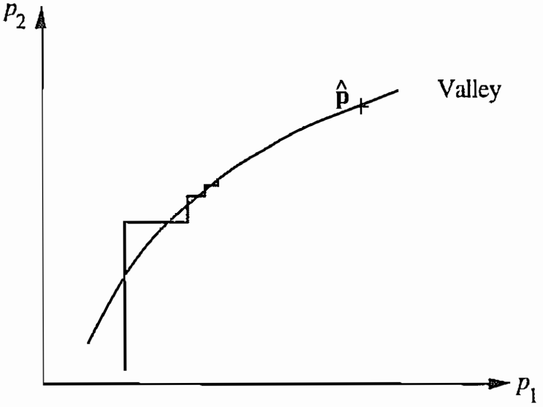


Figure 4.12. Limitations of cyclical one-dimensional searches

The minimization in Step 2 is carried out in the average direction of the steps in Step 1. It therefore allows steps that are better oriented with respect to the valley (Figure 4.13). When the cost is quadratic in $\mathbf{p}$, the method converges in one iteration, *i.e.* $n_\mathrm{p}+1$ one-dimensional minimizations.

Eliminating, at Step 3, the direction that yielded the best results may seem surprising at first sight. The purpose of this policy is to escape the tendency of the $\mathbf{d}_i$'s to become linearly dependent as the iterations proceed. As a matter of fact, the best direction is often very close to that introduced at Step 2. Another solution is to reinitialize the directions $\mathbf{d}_i$ periodically.

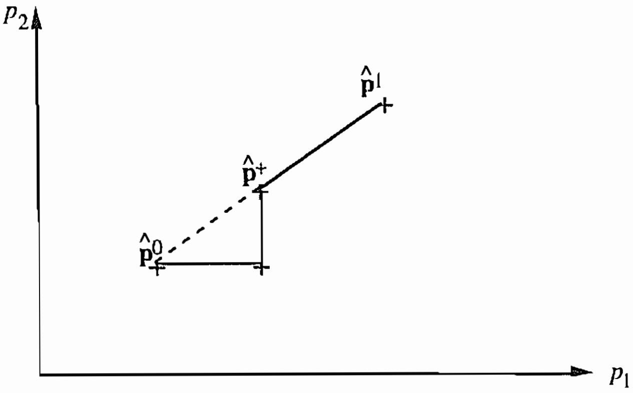


Figure 4.13. Simplest version of Powell's method

A more sophisticated implementation, generating mutually conjugate search directions (see Section 4.3.3.8 for a definition) when applied to a quadratic function, is as follows (Minoux, 1983; Press *et al.*, 1986):

Step 1: Starting from $\mathbf{p}_0 = \hat{\mathbf{p}}^k$, compute $\hat{\mathbf{p}}^{k+}$ by performing $n_\mathrm{p} = \dim \mathbf{p}$ one-dimensional minimizations in linearly independent directions $\mathbf{d}_i$ ($i = 1, \dots, n_\mathrm{p}$):

$$\mathbf{p}_i = \mathbf{p}_{i-1} + \lambda_i \mathbf{d}_i, \quad \text{with} \quad \lambda_i = \arg \min_{\lambda} [j(\mathbf{p}_{i-1} + \lambda \mathbf{d}_i)],$$

$$\hat{\mathbf{p}}^{k+} = \mathbf{p}_{n_\mathrm{p}}.$$

Step 2: Let

$$\Delta = \max_{i=1,\dots,n_\mathrm{p}} [j(\mathbf{p}_{i-1}) - j(\mathbf{p}_i)],$$

$$j_k = j(\hat{\mathbf{p}}^k), \quad j_{k+} = j(\hat{\mathbf{p}}^{k+}) \quad \text{and} \quad j_{k++} = j(2\hat{\mathbf{p}}^{k+} - \hat{\mathbf{p}}^k).$$

If

$$j_{k++} \geq j_k \quad \text{or} \quad (j_k - 2j_{k+} + j_{k++})(j_k - j_{k+} - \Delta)^2 \geq \frac{1}{2} \Delta (j_k - j_{k++})^2,$$

then $\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^{k+}$, increment k by one and go to Step 1. Else, find $\hat{\mathbf{p}}^{k+1}$ by an $(n_\mathrm{p} + 1)$th minimization in the direction

$$\mathbf{d}_{n_\mathrm{p}+1} = \hat{\mathbf{p}}^{k+} - \hat{\mathbf{p}}^k.$$

Step 3: Let

$$\hat{\imath} = \arg \max_{i=1,\dots,n_\mathrm{p}} [j(\mathbf{p}_{i-1}) - j(\mathbf{p}_i)].$$

Remove the direction indexed by $\hat{\imath}$ from the list of search directions, and add $\mathbf{d}_{n_\mathrm{p}+1}$ *at the end* of the list. (The order in which the directions are considered is important.) Increment k by one and go to Step 1.

Again, the initial directions $\mathbf{d}_i$ can be chosen parallel to the axes of parameter space.

When j is continuously differentiable, Powell's method converges towards a local optimizer. The asymptotic behaviour of this version is similar to that of the conjugate-gradient algorithms to be presented in Section 4.3.3.8.

In general, the one-dimensional search technique employed in Powell's method does not use the derivative of the cost with respect to the parameters (Brent, 1973; Press *et al.*, 1986), which does not, however, necessarily mean that it is appropriate for a non-differentiable cost; see Section 4.3.5. When this derivative is available, it is often preferable to analyse the local properties of the cost so as to decide in which direction the one-dimensional search should be performed.

To illustrate the behaviour of various algorithms on the same problems, we shall consider two cost functions. Example 4.10 employs the well known Rosenbrock function. The cost function of Example 4.11 is associated with a least-squares problem to be presented in more detail in Section 4.3.9.1 (Example 4.21). In this problem, the parameters are only locally identifiable, and there are two global minimizers (giving the same value of the cost).

EXAMPLE 4.10

Consider the Rosenbrock cost function

$$j(\mathbf{p}) = 100\,(p_2 - p_1^2)^2 + (1 - p_1)^2,$$

with $\mathbf{p} = (p_1, p_2)^{\mathrm{T}}$. Although it looks trivial, it is actually a rather difficult test case because of its non-convexity, evidenced by the shape of the cost contours plotted in Figure 4.14. The value of the minimizer $\hat{\mathbf{p}} = (1, 1)^{\mathrm{T}}$ is indicated by a cross. All algorithms will be started from $\hat{\mathbf{p}}^0 = (-1,\ 2.5)^{\mathrm{T}}$, indicated on Figure 4.14 by a circle, which corresponds to $j(\hat{\mathbf{p}}^0) = 229$.

Figure 4.14 shows the trajectory followed by the second version of Powell's algorithm, with one-dimensional search by Brent's (derivative-free) algorithm (Press *et al.*, 1986). Optimization is first carried out along the p_1 axis, then proceeds along the valley. After 1168 evaluations of the cost and none of its gradient, the value of the cost is equal to 1.78×10^{-29}. ◊

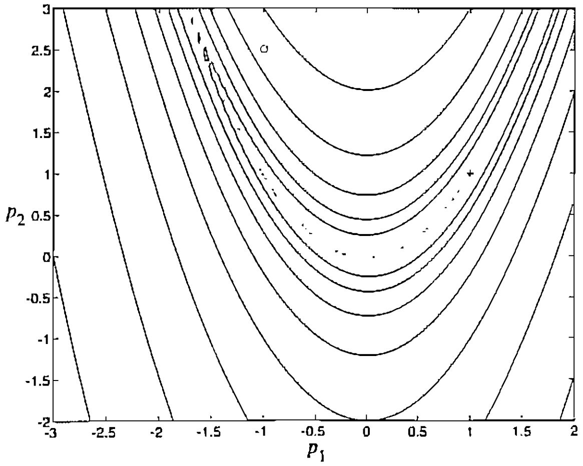


Figure 4.14. Behaviour of Powell's method on Rosenbrock's test function; the initial value of **p** is indicated by a circle, and the minimizer by a cross

EXAMPLE 4.11

Assume now that

$$j(\mathbf{p}) = (p_1 + p_2^2 - 5)^2 + (p_1 + p_2 - 2)^2 + (p_1 + p_2 - 4)^2$$

is to be minimized. All algorithms will be started from $\hat{\mathbf{p}}^0 = (-4,\ 4.9)^{\mathrm{T}}$, which corresponds to $j(\hat{\mathbf{p}}^0) = 236.12$. Figure 4.15 presents the trajectory followed by the same version of Powell's method as in Example 4.10. The first search direction is again along the p_1 axis, so the cost is decreased by moving away from the minimizers. The p_2 axis turns out to be associated with the smallest decrease of the cost, and is therefore frequently used. The minimum at $\hat{\mathbf{p}} = (1, 2)^{\mathrm{T}}$ is reached with four significant digits in 159 evaluations of the cost and none of its gradient. For other values of $\hat{\mathbf{p}}^0$, the method would converge to the other minimizer $\hat{\mathbf{p}} = (4, -1)^{\mathrm{T}}$. ◊

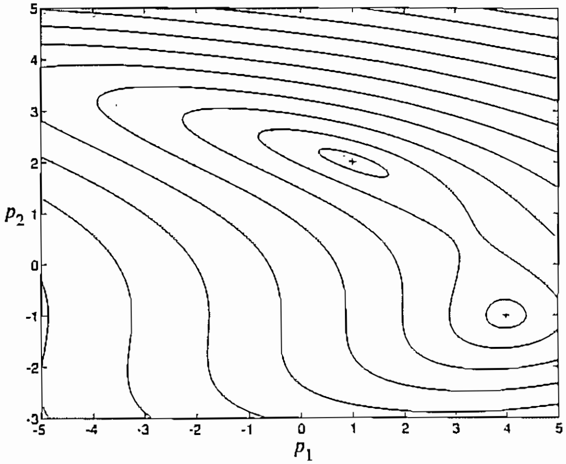


Figure 4.15. Behaviour of Powell's method on Example 4.11; the initial value of $\mathbf{p}$ is indicated by a circle, and the two global minimizers by crosses

REMARK 4.13

The *simplex method* (Nelder and Mead, 1965), not to be confused with its well known namesake in linear programming (Dantzig, 1963), also makes it possible to minimize a function of several variables without using derivatives. It consists of iterative transformations of a simplex in parameter space, *i.e.* of a polytope with dim $\mathbf{p} + 1$ vertices. The transformations are designed to move the vertices towards a minimizer while maintaining a nonzero volume for the simplex. The basic step replaces the vertex associated with the worst value of the cost by its *reflection* with respect to the mean of all other vertices. If the cost at the new vertex turns out to be better than at any other, further *extrapolation* in the same direction is attempted. Otherwise, various *contractions* of the simplex are performed, depending on simple decision rules.

Since the simplex method does not take the local properties of the cost into account, it can be used to optimize a noisy cost, for example when the operating conditions of a system have to be tuned from direct measurements on it, without the use of a mathematical model. See Section 4.4.1 for a short presentation of methods available in this context.

As regards parameter estimation, the simplex method is no match for the methods presented here that use local properties of the cost. It forms, however, an interesting alternative to Powell's method. On Example 4.10, it reaches $j(\hat{\mathbf{p}}) = 3.31 \times 10^{-10}$ in 187 evaluations of the cost and none of its gradient. On Example 4.11, it finds the minimum at $\hat{\mathbf{p}} = (1, 2)^{\mathrm{T}}$ with four significant digits in 96 evaluations of the cost and none of its gradient.

4.3.3 Limited expansions of the cost

If the cost is sufficiently differentiable with respect to $\mathbf{p}$, a limited expansion about the last estimate $\hat{\mathbf{p}}^k$ can be used to compute a *search direction* in parameter space. A one-dimensional search in this direction will then be performed. The (in)efficiency of an algorithm will be measured by the volume of computation needed to get a given reduction in the value of the cost. The computational effort should be balanced between the choice of the search direction and the one-dimensional search; see, in particular, Wolfe's method, described in Section 4.3.3.9. For theoretical analysis of convergence speeds, see, *e.g.*, (Polak, 1971; Minoux, 1983; Polyak, 1987). Let us start with the simplest method.

4.3.3.1 Gradient method

The gradient method, which can be traced back to Fermat at the beginning of the 17th century, is seldom to be recommended. It seems nevertheless worth presenting, because the problems raised by its implementation are shared by many useful methods, and because many more powerful algorithms use it as a building block. It relies on a first-order expansion of the cost about $\hat{\mathbf{p}}^k$:

$$j(\hat{\mathbf{p}}^{k+1}) = j(\hat{\mathbf{p}}^k + \Delta\mathbf{p}) = j(\hat{\mathbf{p}}^k) + \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\Delta\mathbf{p} + o(\|\Delta\mathbf{p}\|),$$

where

$$\mathbf{g}(\hat{\mathbf{p}}^k) = \frac{\partial j}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\hat{\mathbf{p}}^k}$$

is the *gradient* at $\hat{\mathbf{p}}^k$ of the cost, a column vector.

When the displacement $\Delta\mathbf{p}$ is small enough, the resulting variation Δj of the cost satisfies

$$\Delta j = j(\hat{\mathbf{p}}^k + \Delta\mathbf{p}) - j(\hat{\mathbf{p}}^k) \approx \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\Delta\mathbf{p}.$$

It is therefore approximately equal to the scalar product of the gradient of the cost by $\Delta\mathbf{p}$. Since j is to be minimized, Δj should be minimized. For any given $\|\Delta\mathbf{p}\|$, this leads to choosing $\Delta\mathbf{p}$ colinear with the gradient but in the opposite direction, *i.e.*

$$\Delta\mathbf{p} = -\lambda\mathbf{g}(\hat{\mathbf{p}}^k), \quad \text{with} \quad \lambda > 0.$$

The *gradient algorithm* is thus given by

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \lambda\mathbf{g}(\hat{\mathbf{p}}^k).$$

Whenever $\|\mathbf{g}(\hat{\mathbf{p}}^{k+1})\|$ is considered close enough to zero (Section 4.3.7), the algorithm is stopped.

Think of a climber left alone blindfold on top of a mountain. To return to the valley, he must minimize his altitude by changing his position. The gradient policy amounts to sensing the shape of the ground about his feet to find the direction of steepest rise, then starting in the opposite direction. Needless to say, such a short-sighted policy may let the climber down badly!

Choice of step length. In this simple form, the algorithm indicates the direction in which the step should be taken, but not the distance to be covered, since the choice of λ has yet to be made. Three policies can be considered. The first is to fix λ at a *constant value*. The following theorem (Polyak, 1987) then gives an indication how λ should be chosen.

THEOREM

If the cost j is bounded below ($j(\mathbf{p}) \geq j_{\text{opt}} > -\infty$ for any $\mathbf{p}$) and if its gradient with respect to $\mathbf{p}$ is Lipschitz, with Lipschitz constant L:

$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2,$$

then if $0 < \lambda < 2/L$, the gradient algorithm with constant λ will converge to a stationary point ($\mathbf{g}(\hat{\mathbf{p}}^\infty) = \mathbf{0}$) and the cost will decrease monotonically. If, moreover, j satisfies

$$L_\text{m}\mathbf{I}_{n_\text{p}} \leq \frac{\partial^2 j(\mathbf{p})}{\partial\mathbf{p}\partial\mathbf{p}^\text{T}} \leq L_\text{M}\mathbf{I}_{n_\text{p}}, \text{ with } L_\text{m} > 0,$$

for any $\mathbf{p}$, then $\|\hat{\mathbf{p}}^k - \hat{\mathbf{p}}^\infty\|_2 \leq \|\hat{\mathbf{p}}^0 - \hat{\mathbf{p}}^\infty\|_2\, q^k$, with $q = \max\{|1 - \lambda L_\text{m}|, |1 - \lambda L_\text{M}|\}$. The rate of convergence is therefore that of a geometric progression, with common ratio q. The optimal common ratio $q^* = (L_\text{M} - L_\text{m})/(L_\text{M} + L_\text{m})$ is obtained for $\lambda^* = 2/(L_\text{m} + L_\text{M})$. ◊

Unfortunately, even when the gradient of the cost is Lipschitz, the constants L_m and L_M are generally unknown. If λ is too small, the displacements $\Delta\mathbf{p}$ will also be too small, which will slow down convergence. Conversely, if λ gets too large, the first-order approximation will no longer be valid, which may cause divergence of the algorithm. Most often, the value of λ to ensure a decent convergence rate depends on the location of $\hat{\mathbf{p}}^k$. The policy of assigning a constant value to λ is therefore unacceptable.

A second possible policy is to take λ at iteration k as the kth element of a series satisfying (Polyak, 1966)

$$\lambda_k \to 0 \text{ as } k \to \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \lambda_k = \infty.$$

This method will be considered again in Section 4.3.5.1 for the optimization of functions that are not differentiable everywhere. It often exhibits extremely slow convergence (Minoux, 1983).

A third possible policy is to determine λ at each iteration by *one-dimensional search*, usually *via* a quadratic approximation. In *steepest-descent methods*, λ is chosen optimally at each step. Let

$$j_\text{u}(\lambda) = j[\hat{\mathbf{p}}^{k+1}(\lambda)] = j[\hat{\mathbf{p}}^k - \lambda\mathbf{g}(\hat{\mathbf{p}}^k)],$$

and approximate $j_\text{u}(\lambda)$ by a second-order polynomial

$$q(\lambda) = a\lambda^2 + b\lambda + c.$$

Taking advantage of the fact that the gradient of the cost is available at $\hat{\mathbf{p}}^k$, one may compute coefficients a, b and c very simply with a single additional evaluation of the cost. It is enough to require that the polynomial has

— the same value as the cost at the current point $\hat{\mathbf{p}}^k$:

$$c = q(0) = j_{\mathrm{u}}(0) = j(\hat{\mathbf{p}}^k),$$

— the same derivative at $\hat{\mathbf{p}}^k$ in the search direction:

$$b = \frac{\mathrm{d}q(\lambda)}{\mathrm{d}\lambda}\Big|_{\lambda=0} = \frac{\mathrm{d}j_{\mathrm{u}}(\lambda)}{\mathrm{d}\lambda}\Big|_{\lambda=0} = \frac{\partial j(\mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}\Big|_{\mathbf{p}=\hat{\mathbf{p}}^k} \frac{\mathrm{d}\hat{\mathbf{p}}^{k+1}}{\mathrm{d}\lambda}\Big|_{\lambda=0} = -\mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\mathbf{g}(\hat{\mathbf{p}}^k) = -\|\mathbf{g}(\hat{\mathbf{p}}^k)\|_2^2,$$

— the same value at a point λ_1 to be chosen.

When the minimum value of the cost is close to zero, one may for instance take $\lambda_1 = -2c/b$, which amounts to choosing twice the value of λ that would make $j_{\mathrm{u}}(\lambda)$ zero if the first-order approximation $j_{\mathrm{u}}(\lambda) \approx b\lambda + c$ were exact. One then gets

$$a = \frac{1}{\lambda_1^2}\{j[\hat{\mathbf{p}}^k - \lambda_1 \mathbf{g}(\hat{\mathbf{p}}^k)] - b\lambda_1 - c\}.$$

Once the approximating polynomial has thus been determined, the step size $\hat{\lambda}$ is chosen by minimizing $q(\lambda)$. The values of the parameters at the next iteration are then given by

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \hat{\lambda}\mathbf{g}(\hat{\mathbf{p}}^k), \quad \text{with} \quad \hat{\lambda} = -\frac{b}{2a}.$$

A fourth option, efficient and very simple to implement, is to *adapt* λ depending on the present performance. The algorithm is then written as

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \lambda_k \mathbf{g}(\hat{\mathbf{p}}^k),$$

and λ_k is modified at each iteration depending on how the cost evolves. Three cases may arise:

— If $j(\hat{\mathbf{p}}^{k+1}) < j(\hat{\mathbf{p}}^k)$, everything is going fine, and one attempts to accelerate convergence by increasing λ (*e.g.*, $\lambda_{k+1} = 1.5\lambda_k$).

— If $j(\hat{\mathbf{p}}^{k+1}) = j(\hat{\mathbf{p}}^k)$, this most often means that $\Delta\mathbf{p}$ has got so small that the variation of the cost is no longer visible given the accuracy with which the computation is performed. Here too, λ should be increased.

— Finally, if $j(\hat{\mathbf{p}}^{k+1}) > j(\hat{\mathbf{p}}^k)$, the algorithm has gone too far, and the first-order approximation is no longer valid. The step size λ should be reduced (*e.g.*, $\lambda_{k+1} = 0.5\lambda_k$), and $\hat{\mathbf{p}}^{k+1}$ should be replaced by $\hat{\mathbf{p}}^k$.

Properties of the gradient algorithm

PG1: It is simple to implement and has a large convergence domain.

PG2: Usually, the closer $\hat{\mathbf{p}}^k$ is to an optimizer, the slower the convergence, quickly becoming unacceptable (see Figure 4.19)!

PG3: PG1 and PG2 make it well suited to the initial phase of search, when $\hat{\mathbf{p}}^k$ is far from an optimizer.

PG4: At iteration k, the search direction is orthogonal to the hyperplane tangent at $\hat{\mathbf{p}}^k$ to the cost contour at level $j(\hat{\mathbf{p}}^k)$ (locus of all $\mathbf{p}$'s with the same cost $j(\hat{\mathbf{p}}^k)$).

PG5: The path followed by the algorithm is not invariant under a reparametrization, corresponding for instance to a change in the units, as illustrated by the next example.

EXAMPLE 4.12

Assume that the cost to be minimized is

$$j(\mathbf{p}) = \frac{1}{2}(p_1^2 + p_2^2).$$

The cost contours are therefore circles centred on the origin. The search direction, given by

$$-\frac{\partial j}{\partial \mathbf{p}} = -(p_1, \quad p_2)^{\mathrm{T}},$$

is locally orthogonal to the cost contour passing through the current point. It therefore goes through the origin, and a one-dimensional search leads directly to the minimizer (Figure 4.16a). Suppose now that the parametrization is modified, by a mere change of unit, into

$$q_1 = p_1, \quad q_2 = 0.1p_2 .$$

The cost then becomes

$$j(\mathbf{q}) = \frac{1}{2}(q_1^2 + 100q_2^2).$$

The cost contours are now ellipses centred on the origin. The search direction, given by

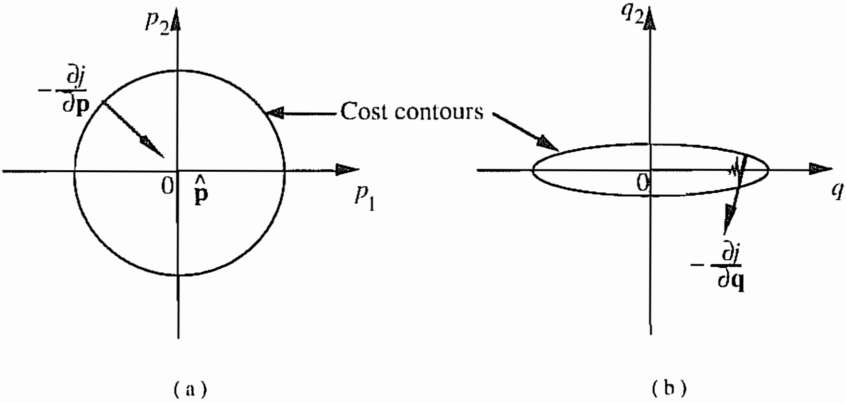
$$-\frac{\partial j}{\partial \mathbf{q}} = -(q_1, \quad 100q_2)^{\mathrm{T}},$$

remains perpendicular to the cost contour, but no longer goes through the origin. Convergence to $\hat{\mathbf{q}}$ will be by the valley, and thus not by the shortest route. The trajectory followed will oscillate (Figure 4.16b). Even with an efficient one-dimensional search, infinitely many iterations will be needed to reach $\hat{\mathbf{q}}$. ◊

PG6: Progression towards a minimizer is thus best when the cost contours are made spherical, *i.e.* the parameters have comparable influences. A suitable choice of scale for the parameters is therefore crucial. Unfortunately, the parameters often have a very unbalanced influence on the cost. For instance, if

$$y_{\mathrm{m}}(t, \mathbf{p}) = \sum_{i=1}^{3} a_i \exp(-\lambda_i t),$$

the influence of the λ_i's on the cost will be much more important than that of the a_i's. This is an additional argument in favour of the method described in Section 4.3.1 (Example 4.9) if a cost quadratic in the output error is to be minimized.



p_2
$-\frac{\partial j}{\partial \mathbf{p}}$
Cost contours
0
$\hat{\mathbf{p}}$
p_1
(a)
q_2
0
q_1
$-\frac{\partial j}{\partial \mathbf{q}}$
(b)

Figure 4.16. Influence of parametrization on the gradient algorithm

PG7: Provided that the one-dimensional search accurately locates minimizers, successive search directions are orthogonal.

PG8: The angle ϕ between the successive search directions is indicative of how the gradient algorithm is behaving, as for many algorithms described in the remainder of this chapter. Good one-dimensional optimization should make $\cos \phi$ close to zero. If $\cos \phi > 0$, the angle between consecutive search directions is acute; no oscillation is taking place, and the gradient method leads to consistent decisions. Conversely, if $\cos \phi < 0$ the successive search directions become antagonistic. One can then drastically improve the algorithm by taking the bisector of ϕ as the one-dimensional search direction, as illustrated by Figure 4.17 (Vignes, 1969; Vignes, Alt and Pichat, 1980). Another solution to avoid oscillating around a valley is to perform a one-dimensional search in the direction $\hat{\mathbf{p}}^k - \hat{\mathbf{p}}^{k-n_\mathrm{p}}$ every n_p iterations (Forsythe, 1968; Luenberger, 1973), which is similar to the basic idea of Powell's method.

PG9: If the cost is inverse multimodal, *i.e.* if it has several minimizers, the gradient method will converge to a local or global minimizer from almost any initial point $\hat{\mathbf{p}}^0$. (If $\hat{\mathbf{p}}^0$ is picked at random in $\mathbb{P}$, the probability of converging to a maximizer or a saddle point is zero, because the initial value must be right on it.) The minimizer towards which convergence occurs depends on the value of $\hat{\mathbf{p}}^0$. The set of all values of $\hat{\mathbf{p}}^0$ from which the algorithm converges to a given minimizer is called the *basin of attraction* of this minimizer.

PG10: If some parameters of the model are unidentifiable, the algorithm will not detect it. It will still converge to an element (depending on $\hat{\mathbf{p}}^0$) of the set of all local and global minimizers.

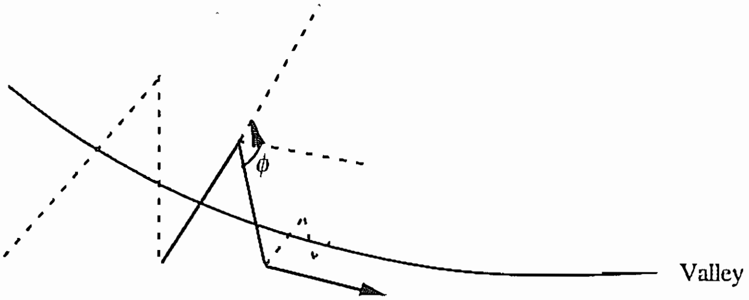


Figure 4.17. Improved search direction when the gradient is oscillating

PG11: Let ψ be the angle between the actual search direction and that suggested by the gradient. Provided that $\cos \psi > 0$, a one-dimensional search can lead to a decrease in the value of the cost, since the gradient is always locally orthogonal to the cost contour (Figure 4.18).

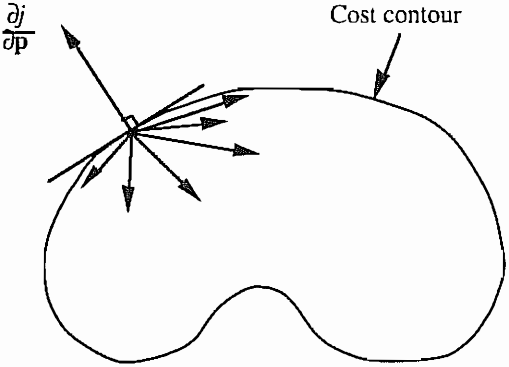


Figure 4.18. Set of all search directions that can locally lead to a decrease in the cost

PG12: The influence of the past is summarized in the state $(\lambda_k, \hat{\mathbf{p}}^k)$ of the algorithm. At each iteration, previous numerical errors only influence the initial point of the rest of the search. There is therefore no accumulation of errors.

PG13: PG11 and PG12 imply that the gradient may be computed rather approximately. As long as the error in the search direction does not exceed $\pi/2$, it remains possible to decrease the cost by one-dimensional search.

PG14: If the cost were to be maximized, the sign of $\Delta\mathbf{p}$ would have to be changed.

PG15: Assume that j is twice continuously differentiable, and that the matrix of its second derivatives with respect to $\mathbf{p}$ is positive definite at $\hat{\mathbf{p}}$, with smallest eigenvalue L_{m} and largest eigenvalue L_{M}. The asymptotic convergence (as $k \rightarrow \infty$) of the gradient algorithm with one-dimensional optimization (steepest descent algorithm) is then linear, and satisfies (Minoux, 1983),

$$\lim_{k\to\infty} \frac{j(\hat{\mathbf{p}}^{k+1}) - j(\hat{\mathbf{p}})}{j(\hat{\mathbf{p}}^{k}) - j(\hat{\mathbf{p}})} \leq \left(\frac{L_M - L_m}{L_M + L_m}\right)^2 .$$

The larger L_M is compared with L_m, the worse the conditioning is and the slower this convergence becomes. Given a starting point $\hat{\mathbf{p}}^0$, the exact asymptotic behaviour of the algorithm (in particular its exact asymptotic rate of convergence) is extremely difficult to predict, even for a quadratic function (Pronzato, Wynn and Zhigljavsky, 1995).

EXAMPLE 4.10 (continued)

Consider again Rosenbrock's test function. Figure 4.19 illustrates the behaviour of the steepest-descent algorithm, for which the step length λ is optimized at each step by Brent's method with derivatives (Press *et al.*, 1986). After a few fast steps, progress along the valley becomes extremely slow; 2095 evaluations of the cost and 1598 evaluations of its gradient painfully lead to a value of the cost equal to 4.44. ◊

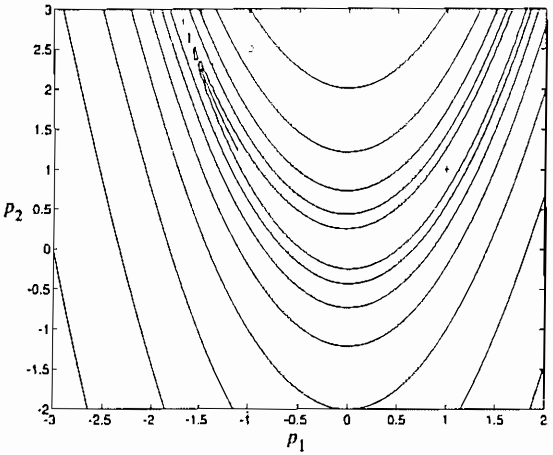


Figure 4.19. Behaviour of the steepest-descent algorithm on Rosenbrock's function; the initial value of $\mathbf{p}$ is indicated by a circle, and the minimizer by a cross

EXAMPLE 4.11 (continued)

Consider again the cost function

$$j(\mathbf{p}) = (p_1 + p_2^2 - 5)^2 + (p_1 + p_2 - 2)^2 + (p_1 + p_2 - 4)^2.$$

The trajectory of the same version of the steepest-descent algorithm as in Example 4.10 is illustrated by Figure 4.20. The successive search directions are approximately orthogonal, which indicates that the one-dimensional minimizations were effective. Each line search is stopped tangentially to a cost contour. After 209 evaluations of the cost, and 156 evaluations of its gradient, the minimum at $\hat{\mathbf{p}} = (1, 2)^{\mathrm{T}}$ is located with an accuracy of four digits. ◊

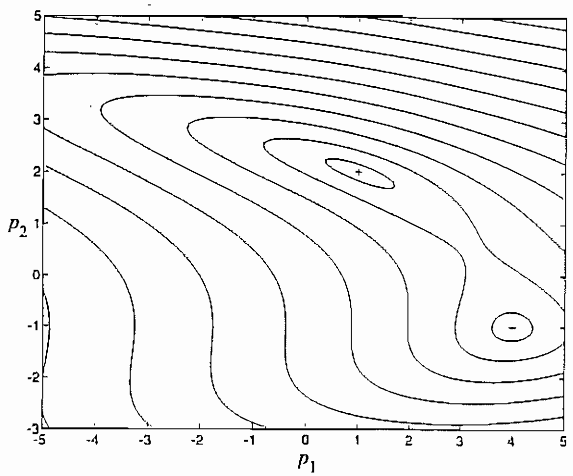


Figure 4.20. Behaviour of the steepest-descent algorithm on Example 4.11; the initial value of **p** is indicated by a circle, and the two global minimizers by crosses

4.3.3.2 Computation of the gradient

The gradient algorithm requires a very large number of evaluations of **g**, which represents a major part of the computation. This is true of all algorithms based on a limited expansion of the cost around the current point $\hat{\mathbf{p}}^k$, as well as some global optimization algorithms, such as the deterministic approach described in Section 4.3.9.3. It may therefore be crucial to make computation of the gradient as fast as possible. The goal of this section is to present some of the techniques that can be used.

Finite differences. Consider the following approximation of the gradient

$$\frac{\partial j(\mathbf{p})}{\partial p_i} \approx \frac{1}{\Delta p_i}[j(\mathbf{p}+\Delta\mathbf{p}_i) - j(\mathbf{p})], \quad i = 1, \dots, n_{\mathrm{p}},$$

where $\Delta\mathbf{p}_i$ is a vector all entries of which are zero except the ith, equal to Δp_i. This approximation requires $n_{\mathrm{p}}+1$ computations of the cost and therefore $n_{\mathrm{p}}+1$ trial steps.

The choice of Δp_i is tricky. If it is too small, the difference of two very similar values of j is evaluated, which may be numerically disastrous since the result may have no significant digit. Conversely, if Δp_i is too large, the quantity evaluated has little resemblance to the derivative, which is less serious since we have seen that a very approximate gradient is often good enough. Section 4.3.7 will present a technique that can be used to estimate the accuracy with which a gradient is evaluated.

Whereas the finite-difference approach always leads to an approximate result, the following techniques make *no approximation*, in principle (although subject to rounding error).

Sensitivity functions. The vector of (first-order) *sensitivity functions* of a scalar error $e(t, \mathbf{p})$ with respect to the parameters is defined as

$$\mathbf{s}_e(t, \mathbf{p}) = \frac{\partial e(t, \mathbf{p})}{\partial \mathbf{p}} = \left[\frac{\partial e(t, \mathbf{p})}{\partial p_1}, \ldots, \frac{\partial e(t, \mathbf{p})}{\partial p_{n_p}}\right]^{\mathrm{T}}.$$

It is usually possible to express the gradient of the cost as a functional of such sensitivity functions. Thus, for instance,

$$j(\mathbf{p}) = \frac{1}{2}\sum_{i=1}^{n_t} w_i[e(t_i, \mathbf{p})]^2 \Rightarrow \frac{\partial j}{\partial \mathbf{p}} = \sum_{i=1}^{n_t} w_i \mathbf{s}_e(t_i, \mathbf{p}) e(t_i, \mathbf{p}).$$

When the error is a vector, its sensitivity functions also enter into the expression for the gradient of the cost. Example 3.4, for instance, suggested using the cost

$$j(\mathbf{p}) = \ln \det \mathbf{M}(\mathbf{p}),$$

where

$$\mathbf{M}(\mathbf{p}) = \frac{1}{n_t}\sum_{i=1}^{n_t} \mathbf{e}(t_i, \mathbf{p})\mathbf{e}^{\mathrm{T}}(t_i, \mathbf{p}).$$

The kth entry of $\mathbf{g}(\mathbf{p})$ is then given (Goodwin and Payne, 1977) by

$$\frac{\partial j}{\partial p_k} = \frac{2}{n_t}\sum_{i=1}^{n_t} \mathbf{e}^{\mathrm{T}}(t_i, \mathbf{p})\mathbf{M}^{-1}(\mathbf{p})\frac{\partial \mathbf{e}(t_i, \mathbf{p})}{\partial p_k}.$$

One possible approach for evaluating the gradient of the cost is therefore to compute the sensitivity functions of the error with respect to each of the parameters. When $e(t, \mathbf{p})$ is an output error,

$$e(t, \mathbf{p}) = y(t) - y_m(t, \mathbf{p}),$$

provided that $y(t)$ does not depend on $\mathbf{p}$, the error sensitivity is related to that of the model output by

$$\mathbf{s}_{y_m}(t, \mathbf{p}) = -\mathbf{s}_e(t, \mathbf{p}).$$

For an LI model described by an nth-order differential equation with known or negligible initial conditions, the sensitivity functions of the model output with respect to the parameters are easy to obtain together with the model output by simulating a differential equation of order $2n$. This will be demonstrated on the following second-order example. Extension to order n and transposition to discrete time are trivial.

EXAMPLE 4.13

Consider the LI model described by

$$\frac{\mathrm{d}^2 y_m}{\mathrm{d}t^2} + p_1\frac{\mathrm{d}y_m}{\mathrm{d}t} + p_2 y_m = p_3\frac{\mathrm{d}u}{\mathrm{d}t} + p_4 u, \quad y_m(0) = 0, \quad \frac{\mathrm{d}y_m}{\mathrm{d}t}\Big|_{t=0} = 0.$$

Let s_i be the sensitivity of the model output with respect to p_i, and assume that the input u does not depend on the model parameters. Differentiating the previous differential equation and initial conditions with respect to p_1, p_2, p_3 and p_4 in turn and changing the order of differentiation gives

$$\begin{aligned}
&\frac{\mathrm{d}^2 s_1}{\mathrm{d}t^2} + p_1 \frac{\mathrm{d}s_1}{\mathrm{d}t} + p_2 s_1 = -\frac{\mathrm{d}y_\mathrm{m}}{\mathrm{d}t}, && s_1(0) = 0, && \frac{\mathrm{d}s_1}{\mathrm{d}t}\Big|_{t=0} = 0,\\
&\frac{\mathrm{d}^2 s_2}{\mathrm{d}t^2} + p_1 \frac{\mathrm{d}s_2}{\mathrm{d}t} + p_2 s_2 = -y_\mathrm{m}, && s_2(0) = 0, && \frac{\mathrm{d}s_2}{\mathrm{d}t}\Big|_{t=0} = 0,\\
&\frac{\mathrm{d}^2 s_3}{\mathrm{d}t^2} + p_1 \frac{\mathrm{d}s_3}{\mathrm{d}t} + p_2 s_3 = \frac{\mathrm{d}u}{\mathrm{d}t}, && s_3(0) = 0, && \frac{\mathrm{d}s_3}{\mathrm{d}t}\Big|_{t=0} = 0,\\
&\frac{\mathrm{d}^2 s_4}{\mathrm{d}t^2} + p_1 \frac{\mathrm{d}s_4}{\mathrm{d}t} + p_2 s_4 = u, && s_4(0) = 0, && \frac{\mathrm{d}s_4}{\mathrm{d}t}\Big|_{t=0} = 0.
\end{aligned}$$

The computation of y_m and the four associated sensitivity functions therefore seems to require use of a 10th-order model, since each equation is of second order. Actually, all these equations have the same homogeneous part, and by using linearity with respect to inputs one can considerably simplify the computation and arrive at the scheme of Figure 4.21, where

$$s_1 = x_3, \quad s_2 = x_4, \quad s_3 = x_1 \quad \text{and} \quad s_4 = x_2.$$

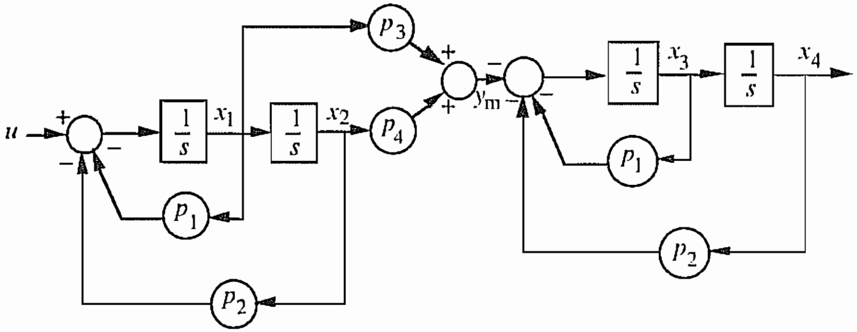


Figure 4.21. Computation of the sensitivity functions of an LI model

Subroutines for simulating continuous-time differential equations usually require that they are provided in state-space form, as

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{p}, \mathbf{u}, t), \quad \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}),$$
$$\mathbf{y}_\mathrm{e} = \mathbf{h}(\mathbf{x}, \mathbf{p}, \mathbf{u}, t).$$

Here, the vector $\mathbf{y}_\mathrm{e}$ of the outputs to be computed consists of the output proper y_m and its sensitivity functions, so $\mathbf{y}_\mathrm{e} = (y_\mathrm{m}, s_1, s_2, s_3, s_4)^\mathrm{T}$. Since the model is LI and has zero initial conditions, the state and observation equations can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \mathbf{A}(\mathbf{p})\mathbf{x} + \mathbf{B}(\mathbf{p})\mathbf{u}, \;\; \mathbf{x}(0) = \mathbf{0},$$
$$\mathbf{y}_{\mathrm{c}} = \mathbf{C}(\mathbf{p})\mathbf{x}.$$

A, **B** and **C** may take various forms, since the state-space representation is not unique. Selecting the outputs of the four integrators as state variables, we can write

$$\mathbf{A}(\mathbf{p}) = \begin{bmatrix} -p_1 & -p_2 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -p_3 & -p_4 & -p_1 & -p_2 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{B}(\mathbf{p}) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ and } \mathbf{C}(\mathbf{p}) = \begin{bmatrix} p_3 & p_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \qquad \Diamond$$

For single-input-single-output time-invariant LI models, it is therefore possible to simplify the computation of sensitivity functions drastically. These results can be extended to multivariable models (Wilkie and Perkins, 1969; Neuman and Sood, 1971) and to time-varying LI models (Neuman and Sood, 1972).

REMARK 4.14

If the effect of unknown initial conditions could not be neglected, less simplification would be possible. ◊

EXAMPLE 4.14

Consider now an ARARMAX model, in the condensed notation introduced in Section 2.4,

$$A(q, \mathbf{p}^*)y(t) = B(q, \mathbf{p}^*)u(t) + \frac{C(q, \mathbf{p}^*)}{D(q, \mathbf{p}^*)}\,\varepsilon(t),$$

which contains ARX, ARMAX and ARARX as special cases. The prediction error can be written as

$$e_{\mathrm{p}}(t, \mathbf{p}) = y(t) - \hat{y}(t|t{-}1, \mathbf{p}) = \frac{D(q, \mathbf{p})}{C(q, \mathbf{p})}\,[A(q, \mathbf{p})y(t) - B(q, \mathbf{p})u(t)].$$

It satisfies

$$e_{\mathrm{p}}(t, \mathbf{p}^*) = \varepsilon(t),$$

and is therefore the error that will appear in the cost obtained by the (conditional) maximum-likelihood method. To compute the gradient of this cost with respect to **p**, one can use the sensitivity of this error with respect to each of the parameters a_i, b_i, c_i and d_i. Assume that the system output y and input u do not depend on the model parameters **p** and that the input-output delay n_{r} is one. Then

$$\frac{\partial}{\partial a_i} e_{\mathrm{p}}(t, \mathbf{p}) = \frac{D(q, \mathbf{p})}{C(q, \mathbf{p})} y(t-i),$$

$$\frac{\partial}{\partial b_i} e_{\mathrm{p}}(t, \mathbf{p}) = -\frac{D(q, \mathbf{p})}{C(q, \mathbf{p})} u(t-i),$$

$$\frac{\partial}{\partial c_i} e_{\mathrm{p}}(t, \mathbf{p}) = -\frac{1}{C(q, \mathbf{p})} e_{\mathrm{p}}(t-i, \mathbf{p}),$$

$$\frac{\partial}{\partial d_i} e_{\mathrm{p}}(t, \mathbf{p}) = \frac{1}{C(q, \mathbf{p})} [A(q, \mathbf{p})y(t-i) - B(q, \mathbf{p})u(t-i)].$$

The sensitivity of the prediction error with respect to $\mathbf{p}$ will be computed by implementing the associated recurrence equations. As in the continuous-time case, these equations have a common autoregressive part. Linearity can therefore again be put to work to simplify computation. If the initial conditions are unknown but the system is stable enough, their effect can be neglected. ◊

REMARK 4.15

The approximate regressor introduced in some methods described in Section 4.2 can be viewed (Ljung and Söderström, 1983) as an approximation of

$$\frac{\partial}{\partial \mathbf{p}} \hat{y}(t|t-1, \mathbf{p}) = -\frac{\partial}{\partial \mathbf{p}} e_{\mathrm{p}}(t, \mathbf{p}). \qquad ◊$$

Sensitivity functions can also be computed for non-LI models; see the interesting review by Rabitz, Kramer and Dacol (1983). As previously, one should differentiate the equations defining the model and its initial conditions with respect to each of the n_{p} parameters in turn. One thus gets n_{p} sets of differential or difference equations with their initial conditions.

Consider, for instance, a state-space equation

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{p}), \quad \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}),$$

where possible dependency on some input $\mathbf{u}$ and time t is omitted to simplify notation. Assume that the model outputs are linear in $\mathbf{x}$:

$$\mathbf{y}_{\mathrm{m}}(t) = \mathbf{C}(\mathbf{p})\mathbf{x}(t).$$

The sensitivity of $\mathbf{y}_{\mathrm{m}}$ with respect to parameter p_i is then given by

$$\frac{\partial \mathbf{y}_{\mathrm{m}}}{\partial p_i} = \frac{\partial \mathbf{C}}{\partial p_i} \mathbf{x} + \mathbf{C} \frac{\partial \mathbf{x}}{\partial p_i}.$$

It can therefore be computed from the sensitivity of $\mathbf{x}$. Differentiating the equations for $\mathbf{x}$ with respect to each parameter in turn, we get:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{s}_i = \frac{\partial \mathbf{f}(\mathbf{x},\mathbf{p})}{\partial \mathbf{x}^{\mathrm{T}}}\mathbf{s}_i + \frac{\partial \mathbf{f}(\mathbf{x},\mathbf{p})}{\partial p_i},\quad \mathbf{s}_i(0) = \frac{\partial \mathbf{x}_0(\mathbf{p})}{\partial p_i},\quad i = 1, \ldots, n_{\mathrm{p}},$$

where

$$\mathbf{s}_i = \frac{\partial \mathbf{x}}{\partial p_i}.$$

The equations satisfied by the state sensitivities $\mathbf{s}_i$ have the following very important properties:

- they are *independent*, so each can be computed in turn, once the state trajectory has been obtained,
- they are *linear* (albeit time-varying since they depend on $\mathbf{x}$ as given by the state equations),
- only the driving term changes with the parameter considered.

This can be exploited to simulate $\mathbf{x}$ and all state sensitivities $\mathbf{s}_i$ more quickly than with the finite-difference approach (Valko and Vajda, 1984; Bilardello *et al.*, 1993), even if computation of the output and n_{p} sensitivity functions requires $n_{\mathrm{p}} + 1$ simulations of non-LI equations, *i.e.* the same number as with the finite-difference approach.

REMARK 4.16

The same type of procedure applies to algebraic-differential systems as considered in Remark 2.2 (Bilardello *et al.*, 1993; Carrillo Le Roux, 1995). Model outputs nonlinear in $\mathbf{x}$ such as

$$y_{\mathrm{m}}(t) = h[\mathbf{x}(t), \mathbf{p}]$$

can also be considered in this framework, by appending to $\mathbf{x}$ a variable $x_{n_{\mathrm{x}}+1}$ that satisfies the algebraic equation

$$x_{n_{\mathrm{x}}+1}(t) - h[\mathbf{x}(t), \mathbf{p}] = 0,$$

so

$$y_{\mathrm{m}}(t, \mathbf{p}) = x_{n_{\mathrm{x}}+1}(t) \qquad \Diamond$$

Adjoint state. For models described by state-space equations, which may be LI or not, adjoint-state techniques borrowed from optimal-control theory make it possible to simplify computation considerably.

Consider a (possibly non-LI) state-space model described by

$$\mathbf{x}(t+1) = \mathbf{f}[\mathbf{x}(t), \mathbf{p}],\ \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}),$$

$$\mathbf{y}_{\mathrm{m}}(t, \mathbf{p}) = \mathbf{h}[\mathbf{x}(t, \mathbf{p}), \mathbf{p}].$$

This model may of course also depend on some input $\mathbf{u}$ and time t, but this dependency is again omitted to simplify notation. Assume that the cost to be optimized is additive, *i.e.* can be written as

$$j(\mathbf{p}) = \sum_{t=0}^{n_{\mathrm{t}}} r_t[\mathbf{x}(t), \mathbf{p}],$$

which is true for most cost functions considered in this book. For a cost quadratic in output error, for instance,

$$r_t[\mathbf{x}(t), \mathbf{p}] = [\mathbf{y}(t) - \mathbf{y}_\mathrm{m}(t, \mathbf{p})]^\mathrm{T}\mathbf{Q}(t)[\mathbf{y}(t) - \mathbf{y}_\mathrm{m}(t, \mathbf{p})]$$

$$= \{\mathbf{y}(t) - \mathbf{h}[\mathbf{x}(t), \mathbf{p}]\}^\mathrm{T}\mathbf{Q}(t)\{\mathbf{y}(t) - \mathbf{h}[\mathbf{x}(t), \mathbf{p}]\}.$$

The additive cost can be transformed into a terminal cost by introducing an additional state variable

$$x^0(t+1) = x^0(t) + r_t[\mathbf{x}(t), \mathbf{p}], \quad \text{with} \quad x^0(0) = 0,$$

so that

$$j(\mathbf{p}) = x^0(n_\mathrm{t}+1).$$

Define the extended state as

$$\mathbf{x}_\mathrm{e}(t) = \begin{bmatrix} x^0(t) \\ \mathbf{x}(t) \\ \mathbf{p} \end{bmatrix}.$$

It satisfies

$$\mathbf{x}_\mathrm{e}(0) = \begin{bmatrix} 0 \\ \mathbf{x}_0(\mathbf{p}) \\ \mathbf{p} \end{bmatrix},$$

and

$$\mathbf{x}_\mathrm{e}(t+1) = \begin{bmatrix} x^0(t) + r_t[\mathbf{x}(t), \mathbf{p}] \\ \mathbf{f}[\mathbf{x}(t), \mathbf{p}] \\ \mathbf{p} \end{bmatrix} = \mathbf{f}_\mathrm{e}[\mathbf{x}_\mathrm{e}(t)].$$

The cost can now be written as

$$j(\mathbf{p}) = [1 \ 0 \ \dots \ 0]\ \mathbf{x}_\mathrm{e}(n_\mathrm{t}+1),$$

which implies that

$$\frac{\partial j}{\partial \mathbf{p}} = \frac{\partial \mathbf{x}_\mathrm{e}^\mathrm{T}(n_\mathrm{t}+1)}{\partial \mathbf{p}} \ \frac{\partial j}{\partial \mathbf{x}_\mathrm{e}(n_\mathrm{t}+1)},$$

with

$$\frac{\partial j}{\partial \mathbf{x}_\mathrm{e}(n_\mathrm{t}+1)} = [1 \ \ 0 \ \dots \ \ 0]^\mathrm{T}.$$

Moreover, since $\mathbf{x}_\mathrm{e}^\mathrm{T}(t+1) = \mathbf{f}_\mathrm{e}^\mathrm{T}[\mathbf{x}_\mathrm{e}(t)]$, the chain rule for the differentiation of composite functions implies that

$$\frac{\partial \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(n_{\mathrm{t}}+1)}{\partial \mathbf{p}} = \frac{\partial \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(n_{\mathrm{t}})}{\partial \mathbf{p}} \frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(n_{\mathrm{t}})]}{\partial \mathbf{x}_{\mathrm{e}}(n_{\mathrm{t}})},$$

$$\frac{\partial \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(n_{\mathrm{t}})}{\partial \mathbf{p}} = \frac{\partial \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(n_{\mathrm{t}}-1)}{\partial \mathbf{p}} \frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(n_{\mathrm{t}}-1)]}{\partial \mathbf{x}_{\mathrm{e}}(n_{\mathrm{t}}-1)},$$

$$\cdots$$

$$\frac{\partial \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(1)}{\partial \mathbf{p}} = \frac{\partial \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(0)}{\partial \mathbf{p}} \frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(0)]}{\partial \mathbf{x}_{\mathrm{e}}(0)}.$$

Globally, one thus gets

$$\frac{\partial j}{\partial \mathbf{p}} = \frac{\partial \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(0)}{\partial \mathbf{p}} \frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(0)]}{\partial \mathbf{x}_{\mathrm{e}}(0)} \frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(1)]}{\partial \mathbf{x}_{\mathrm{e}}(1)} \cdots \frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(n_{\mathrm{t}}-1)]}{\partial \mathbf{x}_{\mathrm{e}}(n_{\mathrm{t}}-1)} \frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(n_{\mathrm{t}})]}{\partial \mathbf{x}_{\mathrm{e}}(n_{\mathrm{t}})} \frac{\partial j}{\partial \mathbf{x}_{\mathrm{e}}(n_{\mathrm{t}}+1)}.$$

The computation associated with this chain-rule differentiation can obviously be carried out from left to right (*direct mode*) or from right to left (*reverse mode*). The reverse mode drastically reduces the number of operations, especially if dim $\mathbf{p}$ is large, because a (dim $\mathbf{x}_{\mathrm{e}}$) vector, the so-called *adjoint state* to the extended state, is propagated instead of a (dim $\mathbf{p}$ × dim $\mathbf{x}_{\mathrm{e}}$) matrix. As will become apparent later, this will however be at the cost of increased memory requirements. Denote the adjoint state by $\mathbf{d}_{\mathrm{x_e}}$. It satisfies the *terminal condition*

$$\mathbf{d}_{\mathrm{x_e}}(n_{\mathrm{t}}+1) = \frac{\partial j}{\partial \mathbf{x}_{\mathrm{e}}(n_{\mathrm{t}}+1)} = [1 \;\; 0 \ldots \;\; 0]^{\mathrm{T}}.$$

Propagating the computation from right to left translates into the *backward-in-time* recurrence equation

$$\mathbf{d}_{\mathrm{x_e}}(t-1) = \frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(t-1)]}{\partial \mathbf{x}_{\mathrm{e}}(t-1)} \mathbf{d}_{\mathrm{x_e}}(t), \;\; t = n_{\mathrm{t}} + 1, \ldots, \;\; 1,$$

which allows $\mathbf{d}_{\mathrm{x_e}}(0)$ to be computed. Note that this requires storage of the trajectories of all extended state variables which $\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}/\partial \mathbf{x}_{\mathrm{e}}$ depends upon, *i.e.* those that appear nonlinearly in $\mathbf{f}_{\mathrm{e}}$. The gradient of the cost is then given by

$$\frac{\partial j}{\partial \mathbf{p}} = \frac{\partial \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(0)}{\partial \mathbf{p}} \mathbf{d}_{\mathrm{x_e}}(0),$$

where

$$\frac{\partial \mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(0)}{\partial \mathbf{p}} = \left[\;\mathbf{0} \quad \frac{\partial \mathbf{x}_{0}^{\mathrm{T}}(\mathbf{p})}{\partial \mathbf{p}} \quad \mathbf{I}_{n_{\mathrm{p}}} \right].$$

REMARKS 4.17

— Forward in time, the evolution equation for the adjoint state becomes

$$\mathbf{d}_{\mathrm{x_e}}(t+1) = \left\{\frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(t)]}{\partial \mathbf{x}_{\mathrm{e}}(t)}\right\}^{-1} \mathbf{d}_{\mathrm{x_e}}(t),$$

while linearization of the state equation around the nominal trajectory yields

$$\delta\mathbf{x}_{\mathrm{e}}(t+1) = \frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(t)]}{\partial \mathbf{x}_{\mathrm{e}}(t)}\,\delta\mathbf{x}_{\mathrm{e}}(t).$$

The scalar product of the linearized and adjoint extended states therefore remains constant along the trajectory (*duality property*)

$$\delta\mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(t+1)\mathbf{d}_{\mathrm{x_e}}(t+1) = \delta\mathbf{x}_{\mathrm{e}}^{\mathrm{T}}(t)\mathbf{d}_{\mathrm{x_e}}(t).$$

This duality can be used to check the computation.

— The method can be implemented without explicitly defining an extended state. As a matter of fact

$$\frac{\partial \mathbf{f}_{\mathrm{e}}^{\mathrm{T}}[\mathbf{x}_{\mathrm{e}}(t)]}{\partial \mathbf{x}_{\mathrm{e}}(t)} = \begin{bmatrix} 1 & \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \\ \dfrac{\partial r_t[\mathbf{x}(t),\mathbf{p}]}{\partial \mathbf{x}(t)} & \dfrac{\partial \mathbf{f}^{\mathrm{T}}[\mathbf{x}(t),\mathbf{p}]}{\partial \mathbf{x}(t)} & \mathbf{0} \\ \dfrac{\partial r_t[\mathbf{x}(t),\mathbf{p}]}{\partial \mathbf{p}} & \dfrac{\partial \mathbf{f}^{\mathrm{T}}[\mathbf{x}(t),\mathbf{p}]}{\partial \mathbf{p}} & \mathbf{I}_{n_{\mathrm{p}}} \end{bmatrix}.$$

The first component of $\mathbf{d}_{\mathrm{x_e}}$ therefore satisfies

$$d_{\mathrm{x}0}(t-1) = d_{\mathrm{x}0}(t), \quad \text{with} \quad d_{\mathrm{x}0}(n_{\mathrm{t}}+1) = 1.$$

The components associated with $\mathbf{x}$ are given by

$$\mathbf{d}_{\mathrm{x}}(t-1) = \frac{\partial \mathbf{f}^{\mathrm{T}}[\mathbf{x}(t),\mathbf{p}]}{\partial \mathbf{x}(t)}\mathbf{d}_{\mathrm{x}}(t) + \frac{\partial r_t[\mathbf{x}(t),\mathbf{p}]}{\partial \mathbf{x}(t)}, \quad \text{with} \quad \mathbf{d}_{\mathrm{x}}(n_{\mathrm{t}}+1) = \mathbf{0},$$

and those corresponding to $\mathbf{p}$ by

$$\mathbf{d}_{\mathrm{p}}(t-1) = \mathbf{d}_{\mathrm{p}}(t) + \frac{\partial \mathbf{f}^{\mathrm{T}}[\mathbf{x}(t),\mathbf{p}]}{\partial \mathbf{p}}\mathbf{d}_{\mathrm{x}}(t) + \frac{\partial r_t[\mathbf{x}(t),\mathbf{p}]}{\partial \mathbf{p}}, \quad \text{with} \quad \mathbf{d}_{\mathrm{p}}(n_{\mathrm{t}}+1) = \mathbf{0}.$$

The gradient of the cost with respect to the parameters is finally obtained as

$$\frac{\partial j}{\partial \mathbf{p}} = \frac{\partial \mathbf{x}_0^{\mathrm{T}}(\mathbf{p})}{\partial \mathbf{p}}\mathbf{d}_{\mathrm{x}}(0) + \mathbf{d}_{\mathrm{p}}(0). \qquad \lozenge$$

In reverse mode, the computation of the gradient proceeds in two phases. The extended state equation is first simulated *forward in time* for the value $\mathbf{p}$ at which the gradient is to be evaluated, which yields the sequence of extended states. The adjoint extended state equation is then simulated *backward in time*, which provides the value of

$\mathbf{d}_{x_c}(0)$ to be used to evaluate the gradient. The gradient is thus computed *exactly* in *two* simulations, up to rounding errors due to finite-precision arithmetic, whatever the number of parameters and even when the state equations are non-LI.

This idea can be transposed to continuous-time models. Simulation of these models usually involves some approximations, and other approximations will take place when the equations describing the evolution of the adjoint state are simulated. Now the computation of gradients by adjoint-state techniques turns out to be rather sensitive to numerical errors, and seemingly small approximations may ruin the final result. It is therefore advisable to make sure, by using a discretized model, that the equations used in the forward and backward directions are subjected to the same approximations. The method presented next pushes this logic even further.

Adjoint code. This type of technique applies to *any* cost j that is computed by a program, provided of course that *j as computed* is differentiable with respect to the parameters $\mathbf{p}$ at the point considered. If, for example, the cost $j(p) = p^2$ was implemented as

$$\text{if } p \neq 1, \text{ then } j(p) = p^2, \text{ else } j(p) = 1,$$

this implementation would not be differentiable at $p = 1$, although the mathematical cost function is.

The sequence of instructions of the program computing the cost forms what will be called the *direct code*, which evaluates $j(\mathbf{p})$ from independent variables ($\mathbf{p}$, $\mathbf{y}^s$, ...), possibly with the use of some intermediate variables, the values of which depend on those of the independent variables.

The adjoint-code approach is especially interesting when the dimension of $\mathbf{p}$ is large, a situation where the use of finite differences or sensitivity functions leads to heavy computation. The basic idea (Gilbert, Le Vey and Masse, 1991; Griewank and Corliss, 1991) is very close to that of the adjoint-state method, and adjoint-code techniques also alternate forward and backward computation. The method to be presented is systematic but usually does not lead to the most concise adjoint code.

Let $\mathbf{v}$ be a vector containing all the variables in the direct code. Any assignment statement of the direct code modifies one component of $\mathbf{v}$, which can be written as

$$v_{\mu(k)} = \phi_k(\{v_i \mid i \in \mathbb{I}_k\}),$$

where $\mu(k)$ is the index of the component of $\mathbf{v}$ that is modified by the kth assignment statement executed, and $\mathbb{I}_k$ is the set of the indices of the components of $\mathbf{v}$ which ϕ_k depends on. The basic idea (Speelpenning, 1980) is to view this instruction as a transformation $\mathbf{\Phi}_k$ of all variables $\mathbf{v}$ of the direct code that leaves all components of $\mathbf{v}$ unchanged but $v_{\mu(k)}$:

$$[\mathbf{\Phi}_k(\mathbf{v})]_i = v_i \,, \quad \forall \ i \neq \mu(k),$$
$$[\mathbf{\Phi}_k(\mathbf{v})]_{\mu(k)} = \phi_k(\{v_i \mid i \in \mathbb{I}_k\}).$$

To distinguish its successive values, $\mathbf{v}$ will be indexed. The initial state of $\mathbf{v}$ will be denoted by $\mathbf{v}(0)$, and its final value by $\mathbf{v}(f)$. The direct code can then be viewed as specifying the evolution of the state of a discrete-time dynamical system, according to

$$\mathbf{v}(k) = \mathbf{\Phi}_k[\mathbf{v}(k-1)], \ \ k = 1, \dots, f.$$

Note that the statement associated with the index k depends on the order in which the statements are executed. Any branching instruction in the direct code therefore requires specific treatment, to be considered later.

In $\mathbf{v}$, the n independent variables will by convention be stored first, starting with the components of $\mathbf{p}$. The dependent variables will then follow, the last component of $\mathbf{v}$ taking the value of the cost $j(\mathbf{p})$ at the end of execution of the direct code. The first n components of the initial state $\mathbf{v}(0)$ are therefore equal to the values of the independent variables for which the cost should be evaluated. All other components of $\mathbf{v}(0)$ may be taken as zero, because the values to be given to the dependent variables result from the execution of the direct code. Once execution has been completed, the value of the cost is the last component of $\mathbf{v}$, *i.e.*

$$j(\mathbf{p}) = [0\ \ldots\ 0\ 1]\ \mathbf{v}(f).$$

The simulation of a discrete-time state equation is thus used to compute a terminal cost, so the adjoint-state method applies. Using the chain rule for differentiation, one can write the gradient of the cost with respect to $\mathbf{p}$ as

$$\frac{\partial j}{\partial \mathbf{p}} = \frac{\partial \mathbf{v}^{\mathrm{T}}(0)}{\partial \mathbf{p}} \frac{\partial \mathbf{\Phi}_1^{\mathrm{T}}}{\partial \mathbf{v}(0)} \frac{\partial \mathbf{\Phi}_2^{\mathrm{T}}}{\partial \mathbf{v}(1)} \cdots \frac{\partial \mathbf{\Phi}_{f-1}^{\mathrm{T}}}{\partial \mathbf{v}(f-2)} \frac{\partial \mathbf{\Phi}_f^{\mathrm{T}}}{\partial \mathbf{v}(f-1)} \frac{\partial j}{\partial \mathbf{v}(f)}.$$

Again, direct and reverse modes can be considered, and reverse mode requires less computation, especially if dim $\mathbf{p}$ is large. The adjoint state $\mathbf{d}$ associated with $\mathbf{v}$ will therefore be initialized with the terminal condition

$$\mathbf{d}(f) = \frac{\partial j}{\partial \mathbf{v}(f)} = [0\ \ldots\ 0\ 1]^{\mathrm{T}},$$

and computed backward according to the formula

$$\mathbf{d}(k-1) = \frac{\partial \mathbf{\Phi}_k^{\mathrm{T}}}{\partial \mathbf{v}(k-1)} \mathbf{d}(k), \quad k = f, \ldots, 1,$$

where $\partial \mathbf{\Phi}_k^{\mathrm{T}}/\partial \mathbf{v}(k-1)$ is an identity matrix, the $\mu(k)$th column of which has been replaced by $\partial \phi_k/\partial \mathbf{v}(k-1)$. Given this specific structure, this recurrence equation translates into

$$d_i(k-1) = d_i(k) + \frac{\partial \phi_k}{\partial v_i(k-1)} d_{\mu(k)}(k) \quad \text{if} \quad i \neq \mu(k),$$

$$d_{\mu(k)}(k-1) = \frac{\partial \phi_k}{\partial v_{\mu(k)}(k-1)} d_{\mu(k)}(k).$$

It makes it possible to compute the initial value $\mathbf{d}(0)$ of the adjoint state. Since

$$\frac{\partial j}{\partial \mathbf{p}} = \frac{\partial \mathbf{v}^{\mathrm{T}}(0)}{\partial \mathbf{p}} \mathbf{d}(0), \quad \text{with} \quad \frac{\partial \mathbf{v}^{\mathrm{T}}(0)}{\partial \mathbf{p}} = [\mathbf{I}_{n_{\mathrm{p}}}\ \ \mathbf{0}],$$

the first n_p components of $\mathbf{d}(0)$ contain the value of the gradient $\mathbf{g}(\mathbf{p})$. Note that the following $n - n_\mathrm{p}$ components of $\mathbf{d}(0)$ contain the values of the partial derivatives of the cost with respect to all other independent variables, such as the data $\mathbf{y}^\mathrm{s}$. These derivatives are therefore *available without any additional computation*. It is not necessary to store all the values taken by $\mathbf{d}(k)$ when k varies from f to 0, since we are only interested in the first n components of $\mathbf{d}(0)$.

The statement

$$v_{\mu(k)} = \phi_k(\{v_i \mid i \in \mathbb{I}_k\})$$

will thus translate into the following adjoint instructions, *in this order*:

$$\textbf{for all } i \in \mathbb{I}_k,\ i \neq \mu(k),\ \textbf{do } \{d_i = d_i + \frac{\partial \phi_k}{\partial v_i} d_{\mu(k)}\};$$

$$d_{\mu(k)} = \frac{\partial \phi_k}{\partial v_{\mu(k)}} d_{\mu(k)}.$$

When the direct variable $v_{\mu(k)}$ is *initialized*, ϕ_k does not depend on $v_{\mu(k)}$, so the last equation reduces to

$$d_{\mu(k)} = 0.$$

When the variable $v_{\mu(k)}$ is *incremented*, by

$$v_{\mu(k)} = v_{\mu(k)} + \psi_k(\mathbf{v}),$$

with ψ_k independent of $v_{\mu(k)}$, it becomes

$$d_{\mu(k)} = d_{\mu(k)}.$$

EXAMPLE 4.15

Assume that the kth assignment statement is

$$cost = cost + [y(t) - y_\mathrm{m}(t)]^2,$$

which sets the dependent variable *cost* to the value

$$\phi_k[cost, y(t), y_\mathrm{m}(t)] = cost + [y(t) - y_\mathrm{m}(t)]^2.$$

Let *dcost*, $dy(t)$ and $dy_\mathrm{m}(t)$ be the dual (adjoint) variables respectively associated with *cost*, $y(t)$ and $y_\mathrm{m}(t)$. Applying the previous formulas, one gets for the adjoint code

$$dy(t) = dy(t) + \frac{\partial \phi_k}{\partial y(t)} dcost,$$

$$dy_\mathrm{m}(t) = dy_\mathrm{m}(t) + \frac{\partial \phi_k}{\partial y_\mathrm{m}(t)} dcost,$$

$$dcost = \frac{\partial \phi_k}{\partial cost} dcost,$$

i.e.

$$dy(t) = dy(t) + 2[y(t) - y_{\mathrm{m}}(t)]dcost,$$

$$dy_{\mathrm{m}}(t) = dy_{\mathrm{m}}(t) - 2[y(t) - y_{\mathrm{m}}(t)]dcost,$$

$$dcost = dcost.$$

The last of these instructions is of course superfluous. ◊

The direct code most often contains branching instructions which must also be dualized. Dualizing an iteration loop (do, for, while...) results in another iteration loop, where the dual instructions are executed in reverse order to the corresponding direct instructions in the direct code. When there are conditional branching instructions, the actual path followed during execution of the direct code must be memorized, so that the dualization of

if (condition C) **then** {code A} **else** {code B}

results in

if (condition C) **then** {adjoint code of A} **else** {adjoint code of B}.

In the adjoint-state method, the evolution of the state was computed forward before computing the evolution of the adjoint state backward. The situation here is analogous: the instructions of the direct code are executed first, then those of the adjoint code, in reverse order. In both cases, one must store the values of those direct variables that appear in nonlinear expressions, in order to allow for the execution of the adjoint computation. These storage requirements are a limiting factor in the complexity of the problems that can be handled in reverse mode; in some cases, the direct mode may turn out to be more feasible.

The procedure can be summarized as follows:

- define **v**, with the variables corresponding to **p** first and the variable corresponding to $j(\mathbf{p})$ last;
- associate an adjoint variable with each component of **v**;
- initialize all adjoint variables to zero, except for the last one, associated with the value of the cost, which is initialized to one;
- dualize the instructions in reverse order of execution, which requires reversing loops;
- execute the direct code, storing the values taken by the direct variables that appear in nonlinear expressions as well as the path followed at conditional branchings;
- execute the adjoint code;
- collect the value of the gradient in the first n components of **d**. The first dim **p** components correspond to the gradient with respect to the parameters, the following ones to the gradient with respect to all other independent variables.

One should be very exacting in these operations, for seemingly minor errors may make the adjoint code useless. Good practice (Talagrand, 1991) is to:

- develop the adjoint code from the direct code, and not from the mathematical adjoint of the direct mathematical equations;

- create an adjoint subroutine for each subroutine of the direct code;
- choose names clearly related to those of the direct variables, labels and subroutines for their adjoint counterparts;
- never modify direct code without propagating the changes in its adjoint.

REMARKS 4.18

- Let $\delta\mathbf{v}(k)$ be the linearized state in the neighbourhood of the nominal trajectory. Its scalar product with the adjoint state remains constant along the trajectory:

$$\mathbf{d}^{\mathrm{T}}(k+1)\delta\mathbf{v}(k+1) = \mathbf{d}^{\mathrm{T}}(k)\delta\mathbf{v}(k).$$

  Advantage can be taken of this duality property to check the validity of an adjoint code, provided that a linearized direct code has been developed to generate the sequence $\{\delta\mathbf{v}\}$.
- Adjoint-code techniques can be extended to the evaluation of higher-order derivatives, such as those needed to evaluate the Hessian of a cost function (Section 4.3.3.3).
- Redundant instructions generated by this systematic approach can be eliminated by using an optimizing compiler. ◊

Summarizing example. Consider a scalar-state model described by

$$x(t+1) = p_1 x(t), \;\; x(0) = p_2,$$

$$y_{\mathrm{m}}(t, \mathbf{p}) = x(t).$$

This is a discrete-time model, but the same type of idea applies to continuous-time models. The parameter vector $\mathbf{p}$ is to be estimated from data $y(t)$ ($t = 1, \ldots, n_t$), by minimizing

$$j(\mathbf{p}) = \sum_{t=1}^{n_t} [y(t) - y_{\mathrm{m}}(t, \mathbf{p})]^2.$$

Let us compute the gradient of this cost at $\hat{\mathbf{p}}^k$ by those methods that provide an exact result. For numerical applications, we shall use

$$\hat{\mathbf{p}}^k = \begin{bmatrix} 0.1 \\ 10 \end{bmatrix}, \quad n_t = 3, \;\; y(1) = 0.5, \;\; y(2) = 0.25, \;\; y(3) = 0.125.$$

The output of the model satisfies $y_{\mathrm{m}}(t, \mathbf{p}) = p_1^t p_2$. It is therefore easy to get an analytical expression for the gradient

$$\frac{\partial j}{\partial p_1}(\mathbf{p}) = -2\sum_{t=1}^{n_t} [y(t) - y_{\mathrm{m}}(t, \mathbf{p})] \frac{\partial y_{\mathrm{m}}(t, \mathbf{p})}{\partial p_1} = -2\sum_{t=1}^{n_t} [y(t) - p_1^t p_2]\, t\, p_1^{t-1} p_2 \,,$$

$$\frac{\partial j}{\partial p_2}(\mathbf{p}) = -2\sum_{t=1}^{n_t} [y(t) - y_m(t, \mathbf{p})]\frac{\partial y_m(t, \mathbf{p})}{\partial p_2} = -2\sum_{t=1}^{n_t} [y(t) - p_1^t p_2] p_1^t \ .$$

For the numerical values chosen,

$$\frac{\partial j}{\partial \mathbf{p}}(\hat{\mathbf{p}}^k) = \begin{bmatrix} 9.331 \\ 0.09677 \end{bmatrix}.$$

The components of the gradient of j also satisfy

$$\frac{\partial j}{\partial p_1}(\mathbf{p}) = -2\sum_{t=1}^{n_t} [y(t) - y_m(t, \mathbf{p})] s_1(t, \mathbf{p}),$$

$$\frac{\partial j}{\partial p_2}(\mathbf{p}) = -2\sum_{t=1}^{n_t} [y(t) - y_m(t, \mathbf{p})] s_2(t, \mathbf{p}),$$

where s_1 and s_2 are the sensitivity functions of the model output with respect to the two parameters:

$$s_1(t, \mathbf{p}) = \frac{\partial y_m(t, \mathbf{p})}{\partial p_1}, \quad s_2(t, \mathbf{p}) = \frac{\partial y_m(t, \mathbf{p})}{\partial p_2}.$$

The equations satisfied by s_1 and s_2 are obtained by differentiating the equations that define the model with respect to $\mathbf{p}$. One gets

$$s_1(t+1, \mathbf{p}) = p_1 s_1(t, \mathbf{p}) + y_m(t, \mathbf{p}), \quad s_1(0, \mathbf{p}) = 0,$$

$$s_2(t+1, \mathbf{p}) = p_1 s_2(t, \mathbf{p}), \quad s_2(0, \mathbf{p}) = 1.$$

For the proposed numerical values, this implies that

$$s_1(1, \hat{\mathbf{p}}^k) = 10, \quad s_1(2, \hat{\mathbf{p}}^k) = 2, \quad s_1(3, \hat{\mathbf{p}}^k) = 0.3,$$

$$s_2(1, \hat{\mathbf{p}}^k) = 0.1, \quad s_2(2, \hat{\mathbf{p}}^k) = 0.01, \quad s_2(3, \hat{\mathbf{p}}^k) = 0.001,$$

and the corresponding value of the gradient is identical to that obtained by analytical differentiation of the cost.

The cost can also be written as

$$j(\mathbf{p}) = \sum_{t=1}^{n_t} r_t[x(t), \mathbf{p}], \quad \text{with} \quad r_t[x(t), \mathbf{p}] = [y(t) - y_m(t, \mathbf{p})]^2.$$

This formulation differs slightly from that considered in the presentation of the adjoint-state method, for the sum starts at $t = 1$ instead of $t = 0$. One could go back to the standard problem by setting $r_0 \equiv 0$, but it is simpler to define the variable x^0 by the recurrence relation

$$x^0(t+1) = x^0(t) + [y(t+1) - y_{\mathrm{m}}(t+1, \mathbf{p})]^2, \quad x^0(0) = 0,$$

which amounts to setting

$$r_t[x(t), \mathbf{p}] = [y(t+1) - p_1 x(t)]^2.$$

As a consequence, $j(\mathbf{p})$ will be given by $x^0(n_t)$ instead of $x^0(n_t+1)$. Define the extended state as $\mathbf{x}_e(t) = [x^0(t), x(t), p_1, p_2]^{\mathrm{T}}$. It satisfies

$$\mathbf{x}_e(t+1) = \begin{bmatrix} x^0(t) + [y(t+1) - p_1 x(t)]^2 \\ p_1 x(t) \\ p_1 \\ p_2 \end{bmatrix} = \mathbf{f}_e[\mathbf{x}_e(t)],$$

$$\mathbf{x}_e(0) = \begin{bmatrix} 0 \\ p_2 \\ p_1 \\ p_2 \end{bmatrix},$$

and

$$j(\mathbf{p}) = [\ 1 \quad 0 \quad 0 \quad 0\ ]\ \mathbf{x}_e(n_t).$$

The adjoint to the extended state

$$\mathbf{d}_{\mathrm{x_e}}(t) = \begin{bmatrix} d_{\mathrm{x^0}}(t) \\ d_{\mathrm{x}}(t) \\ d_{\mathrm{p_1}}(t) \\ d_{\mathrm{p_2}}(t) \end{bmatrix},$$

is given by the backward equation

$$\mathbf{d}_{\mathrm{x_e}}(t-1) = \frac{\partial \mathbf{f}_e^{\mathrm{T}}[\mathbf{x}_e(t-1)]}{\partial \mathbf{x}_e(t-1)}\, \mathbf{d}_{\mathrm{x_e}}(t)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2p_1[y(t) - p_1 x(t-1)] & p_1 & 0 & 0 \\ -2x(t-1)[y(t) - p_1 x(t-1)] & x(t-1) & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{d}_{\mathrm{x_e}}(t),$$

with the terminal condition

$$\mathbf{d}_{\mathbf{x}_\mathrm{e}}(n_\mathrm{t}) = \frac{\partial j}{\partial \mathbf{x}_\mathrm{e}(n_\mathrm{t})} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Taking the last two equations into account, one gets

$$d_{\mathrm{x}0}(t) \equiv 1, \quad d_{\mathrm{p}_2}(t) \equiv 0,$$

$$d_\mathrm{x}(t-1) = p_1 d_\mathrm{x}(t) - 2p_1[y(t) - p_1 x(t-1)], \quad d_\mathrm{x}(n_\mathrm{t}) = 0,$$

$$d_{\mathrm{p}_1}(t-1) = d_{\mathrm{p}_1}(t) + x(t-1)d_\mathrm{x}(t) - 2x(t-1)[y(t) - p_1 x(t-1)], \quad d_{\mathrm{p}_1}(n_\mathrm{t}) = 0,$$

and the gradient of the cost is

$$\frac{\partial j}{\partial \mathbf{p}} = \frac{\partial x_0(\mathbf{p})}{\partial \mathbf{p}} d_\mathrm{x}(0) + \mathbf{d}_\mathrm{p}(0) = \begin{bmatrix} d_{\mathrm{p}_1}(0) \\ d_\mathrm{x}(0) \end{bmatrix}.$$

Evaluating the gradient of j at $\hat{\mathbf{p}}^k$ thus consists of

— a forward phase from $t = 0$ to $t = n_\mathrm{t}$, during which the evolution equation for $\mathbf{x}_\mathrm{e}$, x in practice, is simulated at $\mathbf{p} = \hat{\mathbf{p}}^k$, storing the values of x needed for the backward phase;

— a backward phase from $t = n_\mathrm{t}$ to $t = 0$, during which the evolution equations for d_x and d_{p_1} are simulated at $\mathbf{p} = \hat{\mathbf{p}}^k$, using the values of x stored during the forward phase.

The components of the gradient at $\mathbf{p} = \hat{\mathbf{p}}^k$ are then given by $d_{\mathrm{p}_1}(0)$ and $d_\mathrm{x}(0)$. Applying this procedure with the proposed numerical values gives

$$x(0) = 1, \quad x(1) = 1, \quad x(2) = 0.1, \quad x(3) = 0.01,$$

$$d_\mathrm{x}(3) = 0, \quad d_\mathrm{x}(2) = -0.023, \quad d_\mathrm{x}(1) = -0.0323, \quad d_\mathrm{x}(0) = 0.09677,$$

$$d_{\mathrm{p}_1}(3) = 0, \quad d_{\mathrm{p}_1}(2) = -0.023, \quad d_{\mathrm{p}_1}(1) = -0.346, \quad d_{\mathrm{p}_1}(0) = 9.331.$$

The numerical values found for the gradient are therefore the same as previously.

The direct code can be written as

```
$cost = 0$;
$y_\mathrm{m}(0) = p_2$;
% forward loop
for $k = 1$ to $n_\mathrm{t}$ do {
        $y_\mathrm{m}(k) = p_1 y_\mathrm{m}(k-1)$;
        $cost = cost + [y(k) - y_\mathrm{m}(k)]^2$;
}.
```

The technique presented gives the following adjoint code:

```
% initializing adjoint state variables
$dcost = 1$;
for $k = 1$ to $n_t$ do {
        $dy(k) = 0$;
        $dy_m(k) = 0$;
}
$dp_1 = 0$;
$dp_2 = 0$;
% backward loop
for $k = n_t$ down to 1 do {
        % dualization of the second instruction of the direct loop
        $dy(k) = dy(k) + 2[y(k) - y_m(k)]dcost$;
        $dy_m(k) = dy_m(k) - 2[y(k) - y_m(k)]dcost$;
        $dcost = dcost$;
        % dualization of the first instruction of the direct loop
        $dy_m(k-1) = dy_m(k-1) + p_1 dy_m(k)$;
        $dp_1 = dp_1 + y_m(k-1)dy_m(k)$;
        $dy_m(k) = 0$;
}
% dualization of the instruction that precedes the direct loop
$dp_2 = dp_2 + dy_m(0)$;
$dy_m(0) = 0$;
% the gradient is now given by
$\frac{\partial j}{\partial p_1} = dp_1$;
$\frac{\partial j}{\partial p_2} = dp_2$.
```

This adjoint code is unnecessarily long, but obtained in a systematic manner that can easily be implemented on a computer, contrary to the more concise code that follows

```
$dy_m(n_t+1) = 0$;
for $k = n_t$ down to 1 do {
        $dy_m(k) = -2[y(k) - y_m(k)] + p_1 dy_m(k+1)$;
}
$\frac{\partial j}{\partial p_1} = \sum_{i=1}^{n_t} y_m(i-1)dy_m(i)$;
$\frac{\partial j}{\partial p_2} = p_1 dy_m(1)$.
```

Both provide exact values of the gradient, up to rounding errors introduced by the computer.

Conclusions. A number of methods are available to compute the gradient of a cost with respect to its parameters. The finite-difference method provides an approximate result at a high computational cost. Its simplicity of implementation, however, may lead one to retain it when the optimization algorithm is not too sensitive to errors in the computation of the gradient. When the performance of the optimization algorithm depends critically on the quality of computation of the gradient, as in the conjugate gradient and quasi-Newton algorithms presented below, exact techniques should be preferred. These methods do not imply more complex computation than the finite-difference approach, in fact quite the reverse. The techniques based on sensitivity functions only involve forward computation, in contrast to inverse-mode methods based on adjoint state or code, which alternate forward and backward phases. For optimization problems with a large number of parameters, adjoint-code techniques lead to a drastic reduction in computation, at the expense of an implementation that remains complicated. The continuing development of software automating dualization should remove this difficulty. It should be noted, however, that the storage requirements for the values of all direct variables involved in nonlinear expressions may be very high in large-scale problems.

4.3.3.3 Newton method

This method relies on a second-order expansion of the cost about $\hat{\mathbf{p}}^k$:

$$j(\hat{\mathbf{p}}^{k+1}) = j(\hat{\mathbf{p}}^k + \Delta\mathbf{p}) = j(\hat{\mathbf{p}}^k) + \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\Delta\mathbf{p} + \frac{1}{2}\,\Delta\mathbf{p}^{\mathrm{T}}\mathbf{H}(\hat{\mathbf{p}}^k)\Delta\mathbf{p} + \mathrm{o}(\|\Delta\mathbf{p}\|^2),$$

where $\mathbf{g}(\hat{\mathbf{p}}^k)$ is the gradient at $\hat{\mathbf{p}}^k$ of the cost and $\mathbf{H}(\hat{\mathbf{p}}^k)$ is its *Hessian* at $\hat{\mathbf{p}}^k$, a symmetric $n_{\mathrm{p}} \times n_{\mathrm{p}}$ matrix:

$$\mathbf{H}(\hat{\mathbf{p}}^k) = \frac{\partial^2 j}{\partial\mathbf{p}\partial\mathbf{p}^{\mathrm{T}}}\Big|_{\mathbf{p}=\hat{\mathbf{p}}^k}.$$

Let $\Delta j = j(\hat{\mathbf{p}}^{k+1}) - j(\hat{\mathbf{p}}^k)$. The value of $\Delta\mathbf{p}$ that leads to the largest decrease of the cost, *i.e.* that minimizes Δj, satisfies the necessary optimality condition

$$\frac{\partial \Delta j}{\partial \Delta\mathbf{p}}\Big|_{\Delta\hat{\mathbf{p}}} = \mathbf{0} \approx \mathbf{H}(\hat{\mathbf{p}}^k)\Delta\hat{\mathbf{p}} + \mathbf{g}(\hat{\mathbf{p}}^k),$$

which suggests the step

$$\Delta\hat{\mathbf{p}} = -\mathbf{H}^{-1}(\hat{\mathbf{p}}^k)\mathbf{g}(\hat{\mathbf{p}}^k),$$

provided that the Hessian is invertible, which is assumed. The *Newton algorithm* can then be written as

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \mathbf{H}^{-1}(\hat{\mathbf{p}}^k)\mathbf{g}(\hat{\mathbf{p}}^k),$$

compared with the gradient algorithm $\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \lambda_k\mathbf{g}(\hat{\mathbf{p}}^k)$.

Properties of the Newton algorithm

PN1: The computation required at each step is much heavier that with the gradient method.

PN2: Direction and step size are specified simultaneously, contrary to the gradient method, where the step size remains to be chosen.

PN3: When the cost is quadratic in $\mathbf{p}$ and the Hessian invertible, the method converges to the stationary point of the cost in one step, whatever the initial point $\hat{\mathbf{p}}^0$. Assume for instance that

$$j(\mathbf{p}) = \frac{1}{2}\mathbf{e}^{\mathrm{T}}(\mathbf{p})\mathbf{Q}\mathbf{e}(\mathbf{p}) \text{ and } \mathbf{e}(\mathbf{p}) = \mathbf{y}^{\mathrm{s}} - \mathbf{R}\mathbf{p}.$$

The gradient and Hessian of the cost are then given by

$$\frac{\partial j}{\partial \mathbf{p}} = -\mathbf{R}^{\mathrm{T}}\mathbf{Q}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}\mathbf{p}) \quad \text{and} \quad \frac{\partial^2 j}{\partial \mathbf{p}\partial \mathbf{p}^{\mathrm{T}}} = \mathbf{R}^{\mathrm{T}}\mathbf{Q}\mathbf{R},$$

so

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k + (\mathbf{R}^{\mathrm{T}}\mathbf{Q}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{Q}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}\hat{\mathbf{p}}^k) = \hat{\mathbf{p}}^k + (\mathbf{R}^{\mathrm{T}}\mathbf{Q}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{Q}\mathbf{y}^{\mathrm{s}} - \hat{\mathbf{p}}^k = \hat{\mathbf{p}}_{\mathrm{ls}}.$$

The Newton algorithm therefore becomes the least-squares algorithm.

PN4: Even when the cost is not quadratic in $\mathbf{p}$, convergence is usually very quick when it takes place. Typically, five iterations yield a much better result than a thousand iterations of a gradient algorithm. The improvement gets clearer the more ill-conditioned the problem becomes, with non-spherical cost contours. Unfortunately, this is also when errors in the value of the Hessian may have the worst consequences. If $\mathbf{H}$ is Lipschitz around $\mathbf{p}^*$, convergence of the Newton method is locally quadratic, compared with linear convergence of the gradient method (Minoux, 1983). Asymptotically, the number of significant digits in $\hat{\mathbf{p}}^k$ is doubled at each iteration. This, however, is only valid when $\hat{\mathbf{p}}^k$ is close enough to a local optimizer. Far from this optimizer, the performance may turn out to be worse than that of a gradient method.

PN5: The convergence domain is usually much smaller than with a gradient method. As a matter of fact, nothing guarantees that the size of the step $\Delta\hat{\mathbf{p}}$ given by the Newton method will be small enough for a second-order expansion to remain valid.

PN6: When the method converges, it does so to any point in parameter space where the cost function is stationary, which may be a maximizer, a minimizer or a saddle point. The fact that the cost is to be *minimized* has never been taken into account. Were j to be maximized, the sign of $\Delta\mathbf{p}$ would not have to be changed (compare with PG14). Initialization is therefore especially critical, as illustrated by Figure 4.22.

PN7: If the Hessian is positive definite at $\hat{\mathbf{p}}^k$, which nothing guarantees but which is easily checked, the inverse of the Hessian is positive definite too. The absolute value of the angle between the vectors $\Delta\mathbf{p}$ given by a gradient method and the Newton method is then less than $\pi/2$. According to PG11, one can then reduce the cost by a sufficiently small step in the direction

$$\mathbf{d} = -\mathbf{H}^{-1}(\hat{\mathbf{p}}^k)\mathbf{g}(\hat{\mathbf{p}}^k).$$

The Newton algorithm can therefore be modified by introducing a *relaxation coefficient* λ_k:

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \lambda_k \mathbf{H}^{-1}(\hat{\mathbf{p}}^k)\mathbf{g}(\hat{\mathbf{p}}^k).$$

This coefficient $\lambda_k \leq 1$ is saturated at one when $\hat{\mathbf{p}}^k$ gets close to a minimizer. A simple policy is to reject $\hat{\mathbf{p}}^{k+1}$ and divide λ_k by two as long as the cost does not decrease. One should first make sure that $\mathbf{H}(\hat{\mathbf{p}}^k)$ is positive definite, *e.g.* by checking that all its eigenvalues are positive, so that one does move towards a (possibly local) minimizer.

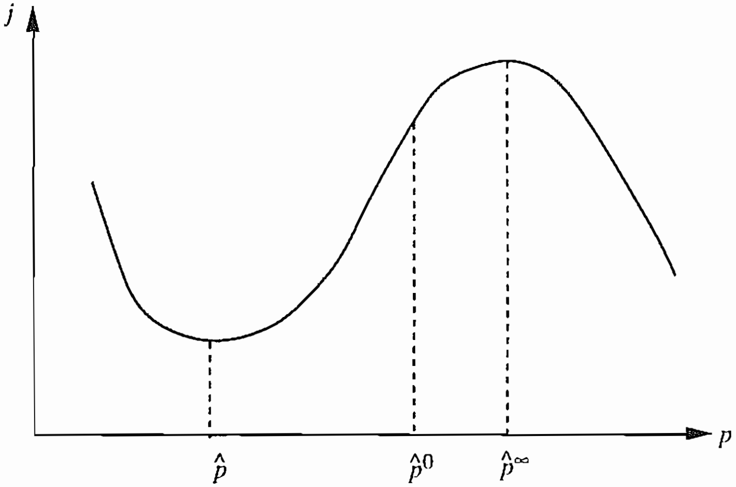


Figure 4.22. From $\hat{p}^0$, the Newton method converges to the locally worst point

PN8: As with the gradient method, the state of the algorithm is summarized by $(\hat{\mathbf{p}}^k, \lambda_k)$. Numerical errors therefore do not accumulate. With the relaxed algorithm, computation of the Hessian may be rather approximate, provided that the result remains positive definite.

PN9: The inverse of the Hessian at the value $\hat{\mathbf{p}}$ of the parameters to which the algorithm converges gives useful information on the uncertainty with which the parameters are estimated (Chapter 5). It is therefore interesting to make the program indicate the value of $\mathbf{H}^{-1}(\hat{\mathbf{p}})$, or at least of its diagonal entries.

Implementation of the Newton method. Rather than using the previous formula, which requires inverting a matrix, it is more economical to solve the following equivalent set of linear equations for $\Delta\mathbf{p}$:

$$\mathbf{H}(\hat{\mathbf{p}}^k)\Delta\mathbf{p} = -\lambda_k \mathbf{g}(\hat{\mathbf{p}}^k).$$

The adjoint-code technique for the computation of gradients can also be employed to compute Hessians exactly. The Hessian is then viewed as the Jacobian matrix associated with the gradient

$$\mathbf{H}(\mathbf{p}) = \frac{\partial^2 j(\mathbf{p})}{\partial \mathbf{p} \partial \mathbf{p}^{\mathrm{T}}} = \frac{\partial \mathbf{g}(\mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}.$$

Let $j_{\mathrm{w}}(\mathbf{p}) = \mathbf{w}^{\mathrm{T}}\mathbf{g}(\mathbf{p})$, where $\mathbf{w}$ is a vector that does not depend on $\mathbf{p}$. The value of the scalar $j_{\mathrm{w}}(\mathbf{p})$ can be computed by a "direct code", which will actually consist of a direct code evaluating $j(\mathbf{p})$ and an adjoint code evaluating $\mathbf{g}(\mathbf{p})$. The "adjoint code" associated with this "direct code" will make it possible to evaluate

$$\frac{\partial}{\partial \mathbf{p}^{\mathrm{T}}} j_{\mathrm{w}}(\mathbf{p}) = \frac{\partial}{\partial \mathbf{p}^{\mathrm{T}}} [\mathbf{w}^{\mathrm{T}}\mathbf{g}(\mathbf{p})].$$

By setting $w_i = \delta_{ik}$, one thus gets the kth row of $\mathbf{H}(\mathbf{p})$. The computation of $\mathbf{H}(\mathbf{p})$ therefore requires n_{p} executions of the "adjoint code". Most of the computation of the "direct code" is to evaluate $\mathbf{g}(\mathbf{p})$, and will be performed only once. Only the last assignment instruction $j_{\mathrm{w}}(\mathbf{p}) = \mathbf{w}^{\mathrm{T}}\mathbf{g}(\mathbf{p})$ will be executed n_{p} times.

Special case of quadratic cost functions. Assume that the cost is

$$j(\mathbf{p}) = \frac{1}{2} \sum_{i=1}^{n_{\mathrm{t}}} w_i [e(t_i, \mathbf{p})]^2.$$

Its gradient satisfies

$$\mathbf{g}(\mathbf{p}) = \sum_{i=1}^{n_{\mathrm{t}}} w_i \frac{\partial e(t_i, \mathbf{p})}{\partial \mathbf{p}} e(t_i, \mathbf{p}),$$

and its Hessian can therefore be written as

$$\mathbf{H}(\mathbf{p}) = \sum_{i=1}^{n_{\mathrm{t}}} w_i \frac{\partial e(t_i, \mathbf{p})}{\partial \mathbf{p}} \frac{\partial e(t_i, \mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}} + \sum_{i=1}^{n_{\mathrm{t}}} w_i \frac{\partial^2 e(t_i, \mathbf{p})}{\partial \mathbf{p} \partial \mathbf{p}^{\mathrm{T}}} e(t_i, \mathbf{p}).$$

In addition to the first-order sensitivity functions already introduced, second-order sensitivity functions now appear:

$$s_{\mathrm{e}ik}(t, \mathbf{p}) = \frac{\partial^2 e(t, \mathbf{p})}{\partial p_i \partial p_k}, \quad i = 1, \ldots, n_{\mathrm{p}}, k = 1, \ldots, n_{\mathrm{p}}.$$

To compute them, one may use similar techniques to those presented for first-order sensitivity functions. One may, for instance, differentiate the equations defining the error with respect to p_i and p_k to get the equations to be solved for $s_{\mathrm{e}ik}$. For an n_{p}-parameter model, it suffices to compute the $n_{\mathrm{p}}(n_{\mathrm{p}} + 1)/2$ entries associated with the upper triangular part of the Hessian, the remainder being obtained by symmetry. One may also use a finite-difference approximation. In both cases, it is easy to see that the computation is much heavier than for the n_{p} first-order sensitivity functions. Nothing even guarantees that the results will be useful, since the Hessian may not be positive definite. Applying adjoint-code techniques also becomes relatively heavy and has the same drawback. This makes the simplification presented in the next section especially interesting.

4.3.3.4 Gauss-Newton method

The Gauss-Newton method, already mentioned in Section 4.2.4, applies when the cost can be expressed as the sum of at least n_p terms quadratic in some error. In this method, the Hessian $\mathbf{H}(\mathbf{p})$ is replaced by a matrix $\mathbf{H}_\mathrm{a}(\mathbf{p})$, obtained by neglecting the term in $\mathbf{H}(\mathbf{p})$ that depends on the second-order sensitivity functions, so that

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^{k} - \lambda_k \mathbf{H}_\mathrm{a}^{-1}(\hat{\mathbf{p}}^{k})\mathbf{g}(\hat{\mathbf{p}}^{k}),$$

where

$$\mathbf{H}_\mathrm{a}(\mathbf{p}) = \sum_{i=1}^{n_\mathrm{t}} w_i \frac{\partial e(t_i, \mathbf{p})}{\partial \mathbf{p}} \frac{\partial e(t_i, \mathbf{p})}{\partial \mathbf{p}^\mathrm{T}} .$$

The computation of $\mathbf{H}_\mathrm{a}$ can therefore be based solely on knowledge of the first-order sensitivity functions, which may already have been used to compute the gradient. Compared with the gradient algorithm, the need to invert a matrix, or rather to solve a linear system of equations, is usually more than compensated by the speeding up of convergence. Approximating the Hessian by $\mathbf{H}_\mathrm{a}$ will be all the more warranted when

— the errors $e(t_i, \mathbf{p})$ are small,
— these errors are little correlated with the second-order sensitivity functions,
— the second-order derivatives of the errors with respect to the parameters remain small, *i.e.* the influence of the parameters on the errors is approximately affine.

Even when these conditions are not satisfied, and provided that the model is locally identifiable under the experimental conditions chosen, the approximate Hessian $\mathbf{H}_\mathrm{a}$ is positive definite, although the Hessian $\mathbf{H}$ is not necessarily so. The matrix $\mathbf{H}_\mathrm{a}^{-1}$ is then positive definite too, so the absolute value of the angle between the search directions suggested by the gradient and Gauss-Newton methods is less than $\pi/2$. It will therefore always be possible to find some positive λ_k that ensures a decrease in the cost. Remember that this was not so for the Newton algorithm.

Since the search direction forms an acute angle with that suggested by the gradient method, the Gauss-Newton algorithm can also be applied to non-quadratric cost functions, provided that a quadratic approximation of the cost is used to compute $\mathbf{H}_\mathrm{a}$.

EXAMPLE 4.10 (continued)

Figure 4.23 illustrates the minimization of Rosenbrock's test function using the Gauss-Newton algorithm. Line searches are performed with the Wolfe method, to be presented in Section 4.3.3.9. The search directions differ from those of a gradient algorithm and no longer start orthogonally to cost contours. After only 19 evaluations of the cost and of its gradient, the cost is down to 1.97×10^{-31}. ◊

EXAMPLE 4.11 (continued)

Consider again the cost function

$$j(\mathbf{p}) = (p_1 + p_2^2 - 5)^2 + (p_1 + p_2 - 2)^2 + (p_1 + p_2 - 4)^2.$$

The trajectory of the same version of the Gauss-Newton algorithm as in Example 4.10 is illustrated by Figure 4.24. The minimum at $\hat{\mathbf{p}} = (1, 2)^{\mathrm{T}}$ is found with four significant digits after only 9 evaluations of the cost, and 13 evaluations of its gradient.◊

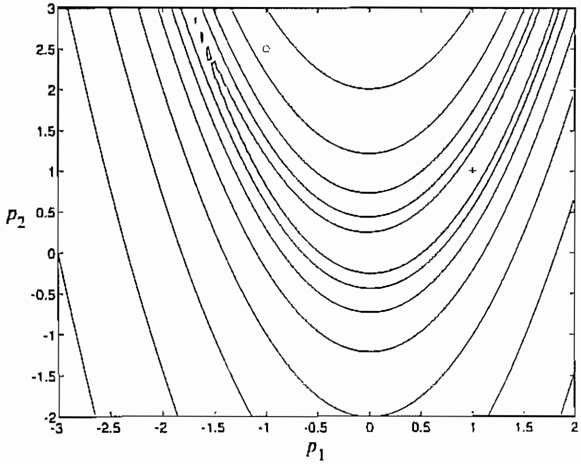


Figure 4.23. Behaviour of the Gauss-Newton algorithm on Rosenbrock's test function; the initial value of **p** is indicated by a circle, and the minimizer by a cross

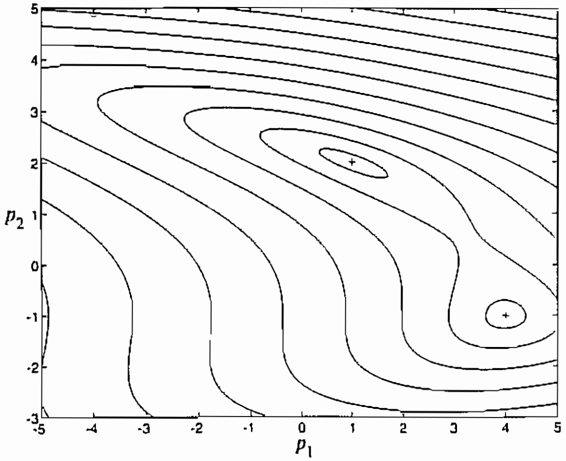


Figure 4.24. Behaviour of the Gauss-Newton algorithm on Example 4.11; the initial value of **p** is indicated by a circle, and the two global minimizers by crosses

4.3.3.5 Levenberg-Marquardt method

The non-relaxed Newton and Gauss-Newton methods require solution for $\Delta\mathbf{p}$ of the linear system of equations

$$\mathbf{H}(\hat{\mathbf{p}}^k)\Delta\mathbf{p} = -\mathbf{g}(\hat{\mathbf{p}}^k),$$

where **H** is either the Hessian or some approximation of it. The Levenberg (1944) and Marquardt (1963) method replaces this equation by

$$[\mathbf{H}(\hat{\mathbf{p}}^k) + \mu_k\mathbf{I}]\Delta\mathbf{p} = -\mathbf{g}(\hat{\mathbf{p}}^k), \quad \text{with} \quad \mu_k \geq 0,$$

which amounts to adding a quadratic penalty function $(\mu_k/2)\|\mathbf{p} - \hat{\mathbf{p}}^k\|_2^2$ to the cost $j(\mathbf{p})$.

When $\mu_k = 0$, one gets a non-relaxed Newton or Gauss-Newton iteration. When, on the other hand, μ_k tends to infinity, the iteration tends to a gradient iteration with a step size $\lambda_k = 1/\mu_k$ that tends to zero. At each iteration, one can therefore perform a one-dimensional search on μ_k. Most often, μ_k is merely adapted, with a policy similar to that presented for the gradient method, *e.g.*

— if $j(\hat{\mathbf{p}}^{k+1}) \leq j(\hat{\mathbf{p}}^k)$, then divide μ_k by 10, (everything is going well, move towards non-relaxed Newton or Gauss-Newton);
— else reject $\hat{\mathbf{p}}^{k+1}$ and multiply μ_k by 10 (the method is diverging, move towards gradient and shorten step size).

Whatever the policy chosen, μ_k should not be allowed to tend to zero, to guard against numerical singularity of **H**. The Levenberg-Marquardt method then regularizes ill-posed problems.

Figure 4.25 illustrates the behaviour of the Levenberg-Marquardt method far from the optimum. Here, the choice $\mu_k = 0$ would be very bad. Conversely, close to the optimum, the cost tends to become quadratic in the parameters, so $\mu_k = 0$ becomes a much better choice.

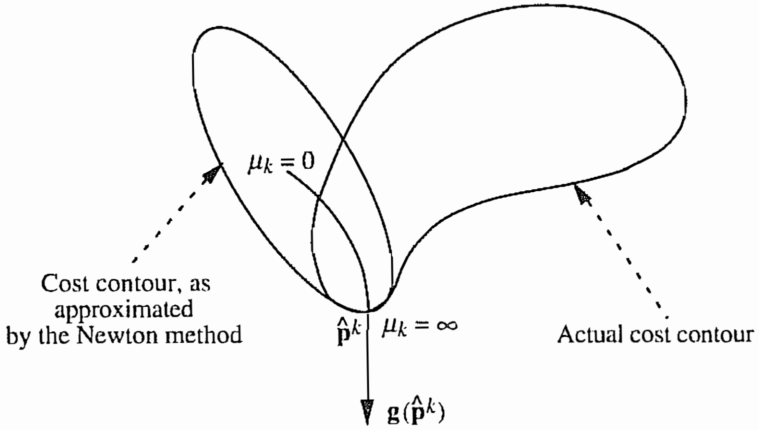

$\mu_k = 0$
Cost contour, as approximated by the Newton method
$\hat{\mathbf{p}}^k$
$\mu_k = \infty$
Actual cost contour
$\mathbf{g}(\hat{\mathbf{p}}^k)$

Figure 4.25. Levenberg-Marquardt method far from the optimum

For any value of μ_k, the computation of $\Delta\mathbf{p}$ requires solution of a set of n_p equations in n_p unknowns. A preliminary diagonalization of $\mathbf{H}(\hat{\mathbf{p}}^k)$, however, makes it possible to write

$$\mathbf{H}(\hat{\mathbf{p}}^k) = \mathbf{T\Lambda T}^\mathrm{T}, \quad \text{with} \quad \mathbf{TT}^\mathrm{T} = \mathbf{I}_{n_\mathrm{p}} \quad \text{and} \quad \mathbf{\Lambda} = \operatorname{diag}\,\{\lambda_i, i = 1, \dots, n_\mathrm{p}\},$$

so

$$\Delta\mathbf{p} = -\,\mathbf{T}\operatorname{diag}\,\{(\lambda_i + \mu_k)^{-1}, i = 1, \dots, n_\mathrm{p}\}\,\mathbf{T}^\mathrm{T}\mathbf{g}(\hat{\mathbf{p}}^k).$$

It is thus possible to replace each new solution of a linear system by a mere matrix multiplication.

4.3.3.6 Quasi-Newton methods

The aim of this family of algorithms is to combine the advantages of the gradient and Newton methods. The idea is to generate, at each iteration, a matrix $\mathbf{M}_k$ that tends to the *inverse* of the Hessian, without ever inverting a matrix (Fletcher and Powell, 1963).

The cost is taken to be a quadratic function of $\mathbf{p}$, so

$$j(\mathbf{p}) = j(\hat{\mathbf{p}}^k) + \mathbf{g}^\mathrm{T}(\hat{\mathbf{p}}^k)(\mathbf{p} - \hat{\mathbf{p}}^k) + \frac{1}{2}(\mathbf{p} - \hat{\mathbf{p}}^k)^\mathrm{T}\mathbf{H}(\mathbf{p} - \hat{\mathbf{p}}^k),$$

where $\mathbf{g}(\hat{\mathbf{p}}^k)$ and $\mathbf{H}$ (assumed symmetric and positive definite) are the gradient and Hessian evaluated at $\hat{\mathbf{p}}^k$. Since the cost is assumed to be quadratic in $\mathbf{p}$, the Hessian does not depend on $\hat{\mathbf{p}}^k$. The gradient satisfies

$$\mathbf{g}(\hat{\mathbf{p}}^{k+1}) = \mathbf{g}(\hat{\mathbf{p}}^k) + \mathbf{H}(\hat{\mathbf{p}}^{k+1} - \hat{\mathbf{p}}^k).$$

Its variation between two iterations is therefore related to the variation of the parameters by

$$\Delta\mathbf{g}_k = \mathbf{H}\Delta\mathbf{p}_k,$$

where

$$\Delta\mathbf{g}_k = \mathbf{g}(\hat{\mathbf{p}}^{k+1}) - \mathbf{g}(\hat{\mathbf{p}}^k) \quad \text{and} \quad \Delta\mathbf{p}_k = \hat{\mathbf{p}}^{k+1} - \hat{\mathbf{p}}^k.$$

Starting from some initial point $\hat{\mathbf{p}}^k$ and some initial (symmetric) approximation $\mathbf{M}_k$ of the inverse of the Hessian, let us see how to compute $\hat{\mathbf{p}}^{k+1}$ and $\mathbf{M}_{k+1}$. To calculate the step in parameter space, a relaxed Newton method is used, where $\mathbf{M}_k$ is substituted for the inverse of the Hessian:

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \lambda_k\mathbf{M}_k\mathbf{g}(\hat{\mathbf{p}}^k).$$

The relaxation parameter λ_k is chosen by one-dimensional optimization, usually by polynomial interpolation.

The approximation of the inverse of the Hessian is updated by

$$\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{C}_k,$$

where $\mathbf{C}_k$ is a symmetric correction matrix. We shall calculate here a rank-one correction, before describing more efficient rank-two corrections. If $\mathbf{M}_{k+1}$ were the inverse of $\mathbf{H}$, one would have

$$\mathbf{M}_{k+1}\Delta\mathbf{g}_k = \Delta\mathbf{p}_k,$$

i.e.

$$(\mathbf{M}_k + \mathbf{C}_k)\,\Delta\mathbf{g}_k = \Delta\mathbf{p}_k.$$

This implies that

$$\mathbf{C}_k\Delta\mathbf{g}_k = \Delta\mathbf{p}_k - \mathbf{M}_k\Delta\mathbf{g}_k = \frac{(\Delta\mathbf{p}_k - \mathbf{M}_k\Delta\mathbf{g}_k)(\Delta\mathbf{p}_k - \mathbf{M}_k\Delta\mathbf{g}_k)^{\mathrm{T}}}{(\Delta\mathbf{p}_k - \mathbf{M}_k\Delta\mathbf{g}_k)^{\mathrm{T}}\Delta\mathbf{g}_k}\,\Delta\mathbf{g}_k.$$

Identifying the coefficients of $\Delta\mathbf{g}_k$ gives

$$\mathbf{C}_k = \frac{(\Delta\mathbf{p}_k - \mathbf{M}_k\Delta\mathbf{g}_k)(\Delta\mathbf{p}_k - \mathbf{M}_k\Delta\mathbf{g}_k)^{\mathrm{T}}}{(\Delta\mathbf{p}_k - \mathbf{M}_k\Delta\mathbf{g}_k)^{\mathrm{T}}\Delta\mathbf{g}_k},$$

which is symmetric and rank-one.

The method is initialized by setting $\mathbf{M}_0 = \mathbf{I}$, so the first iteration is a gradient step. As the iterations proceed, $\mathbf{M}_k$ resembles the inverse of the Hessian more and more, and the method behaves more and more as a relaxed Newton method. This is appropriate, given the properties of these two algorithms.

In practice, a rank-two correction is usually preferred. Various correction formulas are available; see, *e.g.*, Minoux (1983) or Polyak (1987). The *Davidon-Fletcher-Powell* (DFP) method has

$$\mathbf{C}_k = \frac{\Delta\mathbf{p}_k\Delta\mathbf{p}_k^{\mathrm{T}}}{\Delta\mathbf{p}_k^{\mathrm{T}}\Delta\mathbf{g}_k} - \frac{\mathbf{M}_k\Delta\mathbf{g}_k\Delta\mathbf{g}_k^{\mathrm{T}}\mathbf{M}_k}{\Delta\mathbf{g}_k^{\mathrm{T}}\mathbf{M}_k\Delta\mathbf{g}_k},$$

and the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) method has

$$\mathbf{C}_k = \left(1 + \frac{\Delta\mathbf{g}_k^{\mathrm{T}}\mathbf{M}_k\Delta\mathbf{g}_k}{\Delta\mathbf{p}_k^{\mathrm{T}}\Delta\mathbf{g}_k}\right)\frac{\Delta\mathbf{p}_k\Delta\mathbf{p}_k^{\mathrm{T}}}{\Delta\mathbf{p}_k^{\mathrm{T}}\Delta\mathbf{g}_k} - \frac{\Delta\mathbf{p}_k\Delta\mathbf{g}_k^{\mathrm{T}}\mathbf{M}_k + \mathbf{M}_k\Delta\mathbf{g}_k\Delta\mathbf{p}_k^{\mathrm{T}}}{\Delta\mathbf{p}_k^{\mathrm{T}}\Delta\mathbf{g}_k}.$$

BFGS seems to be less sensitive than DFP to errors incurred during one-dimensional searches.

Properties of quasi-Newton algorithms

PQN1: Each iteration only uses values of the cost and its gradient. The computation is simple, not requiring solution of a set of linear equations in n_{p} unknowns.

PQN2: If the cost is quadratic in $\mathbf{p}$, all matrices $\mathbf{M}_k$ are positive definite and the algorithm converges to the optimum in n_{p} iterations. Under the same conditions, the Newton algorithm would converge in a single iteration, but would require solution of a linear set of equations in n_{p} unknowns.

PQN3: Because of numerical rounding, $\mathbf{M}_k$ may become singular; it is then reset to $\mathbf{I}$, which returns the search momentarily to a gradient algorithm. This restarting procedure may even be imposed every n_p iterations, to avoid having to check that $\mathbf{M}_k$ is positive definite.

PQN4: Since the final value of $\mathbf{M}_k$ approximates the Hessian at the optimum, it provides useful information on the uncertainty in the estimated parameters (Section 5.3).

PQN5: $\mathbf{M}_k$ is obtained by accumulating information gathered in all previous iterations, which makes the method more sensitive to numerical errors than the Gauss-Newton method. The gradient should therefore be evaluated with particular care.

PQN6: Whatever the initial value of $\mathbf{p}$, the method converges to a (possibly local) minimizer of the cost.

PQN7: Convergence is slow at first, then speeds up when $\hat{\mathbf{p}}^k$ gets close enough to the local optimizer for the quadratic approximation to become valid and for $\mathbf{M}_k$ to resemble $\mathbf{H}^{-1}$. Asymptotically, provided that the cost is twice continuously differentiable and the Hessian is positive definite at the optimum, most quasi-Newton methods have superlinear convergence, *i.e.* quicker than that of the gradient method. If, moreover, the Hessian satisfies a Lipschitz condition in the neighbourhood of the optimum, convergence becomes quadratic, as with the Newton method (Minoux, 1983).

PQN8: This class of methods can be seen as gradient methods in a metric iteratively transformed to try and make the cost contours spherical. This is why quasi-Newton methods are also called *variable-metric* methods.

Quasi-Newton methods are implemented in all major scientific software libraries. Refrain from attempting to code them once more! The simplest way to see whether they are useful for a given problem is to try them. The only items of information needed are the rules for evaluating j and its gradient with respect to the parameters, *i.e.* what would be needed to apply the gradient method. Variants are also available where the gradient is evaluated by finite differences, so that the user need only provide the subroutine that computes the cost. For reasons alluded to in PQN5, better performance will however be obtained when the gradient is evaluated exactly.

EXAMPLE 4.10 (continued)

Figure 4.26 illustrates the optimization of Rosenbrock's test function by a quasi-Newton method (BFGS). Line searches are performed with Wolfe's method, to be presented in Section 4.3.3.9. After 88 evaluations of the cost and 102 evaluations of its gradient, the cost reaches the value 2.46×10^{-12}. ◊

EXAMPLE 4.11 (continued)

Consider again the cost function

$$j(\mathbf{p}) = (p_1 + p_2^2 - 5)^2 + (p_1 + p_2 - 2)^2 + (p_1 + p_2 - 4)^2.$$

The trajectory of the same version of BFGS as in Example 4.10 is illustrated by Figure 4.27.

The first line search corresponds to a gradient step, orthogonal to the cost contour from which it originates. As the iterations proceed, the approximation of the inverse of the Hessian improves and so do the search directions, which get markedly different from those of a gradient search. The minimum at $\hat{\mathbf{p}} = (1, 2)^{\mathrm{T}}$ is reached with four significant digits after 28 evaluations of the cost, and 39 evaluations of its gradient. ◊

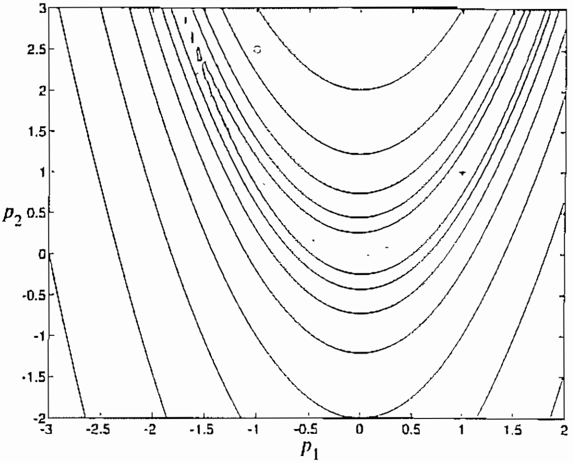


Figure 4.26. Behaviour of a quasi-Newton method (BFGS) on Rosenbrock's test function; the initial value for **p** is indicated by a circle, and the minimizer by a cross

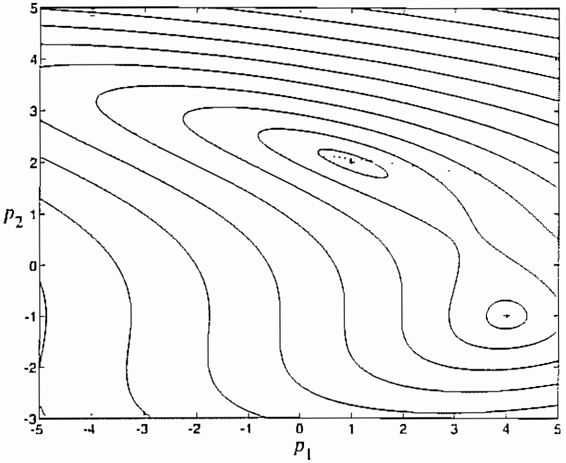


Figure 4.27. Behaviour of BFGS on Example 4.11; the initial value of **p** is indicated by a circle, and the two global minimizers by crosses

4.3.3.7 Heavy-ball method

Except for the quasi-Newton, Powell and Vignes methods, the only information on previous iterations used by the local methods presented so far is the estimate $\hat{\mathbf{p}}^k$ of the parameters, and they are called *one-step* methods. Better use of this information should of course yield a more effective method. Consider, for example, the *two-step* method known as the heavy-ball method (Polyak, 1987),

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \alpha_k \mathbf{g}(\hat{\mathbf{p}}^k) + \beta_k(\hat{\mathbf{p}}^k - \hat{\mathbf{p}}^{k-1}),$$

which reduces to the gradient method for $\beta_k = 0$. It can be viewed as a difference equation describing the motion of a ball in a gravitational field corresponding to the cost in the presence of viscous friction. The loss of energy due to the friction drives the ball to an equilibrium point, which is a local minimizer of j. The term $\beta_k(\hat{\mathbf{p}}^k - \hat{\mathbf{p}}^{k-1})$ may accelerate convergence by damping oscillations perpendicular to the steepest direction. It may also decrease the probability that the algorithm will get trapped in a shallow local minimum. As with the gradient method, the optimal choice of the tuning parameters, here α and β, depends on the cost function and cannot be found *a priori*. To make this approach practicable, the optimal values of α and β should therefore be computed at each iteration, *i.e.*

$$\alpha_k, \beta_k = \arg\min_{\alpha,\beta} j[\hat{\mathbf{p}}^k - \alpha \mathbf{g}(\hat{\mathbf{p}}^k) + \beta(\hat{\mathbf{p}}^k - \hat{\mathbf{p}}^{k-1})].$$

This can be done analytically for quadratic costs, and the heavy-ball method then becomes equivalent to a conjugate-gradient method.

4.3.3.8 Conjugate-gradient methods

As quasi-Newton methods, this family of methods (Fletcher and Reeves, 1964) takes the cost to be a quadratic function of **p**

$$j(\mathbf{p}) = j(\hat{\mathbf{p}}^k) + \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)(\mathbf{p} - \hat{\mathbf{p}}^k) + \frac{1}{2}(\mathbf{p} - \hat{\mathbf{p}}^k)^{\mathrm{T}}\mathbf{H}(\mathbf{p} - \hat{\mathbf{p}}^k),$$

which may also be written as

$$j(\mathbf{p}) = c + \mathbf{b}^{\mathrm{T}}\mathbf{p} + \frac{1}{2}\mathbf{p}^{\mathrm{T}}\mathbf{H}\mathbf{p},$$

with **H** assumed symmetric and positive definite. Let

$$\Delta\mathbf{p}_k = \hat{\mathbf{p}}^{k+1} - \hat{\mathbf{p}}^k,$$

$$\mathbf{g}_k = \mathbf{g}(\hat{\mathbf{p}}^k).$$

If $\hat{\mathbf{p}}^{k+1}$ has been obtained from $\hat{\mathbf{p}}^k$ by one-dimensional minimization along $\mathbf{d}_k$, then

$$\Delta \mathbf{p}_k = \lambda_k \mathbf{d}_k,$$

$$\mathbf{g}_{k+1}^{\mathrm{T}} \mathbf{d}_k = 0.$$

If the next search direction were given by the Newton method, it would satisfy

$$\mathbf{d}_{k+1} = -\mathbf{H}^{-1}\mathbf{g}_{k+1},$$

that is

$$\mathbf{g}_{k+1} = -\mathbf{H}\mathbf{d}_{k+1}.$$

This suggests choosing successive search directions that satisfy

$$\mathbf{d}_{k+1}^{\mathrm{T}} \mathbf{H} \mathbf{d}_k = 0.$$

Such directions are said to be *conjugate* with respect to the Hessian **H** (Figure 4.28).

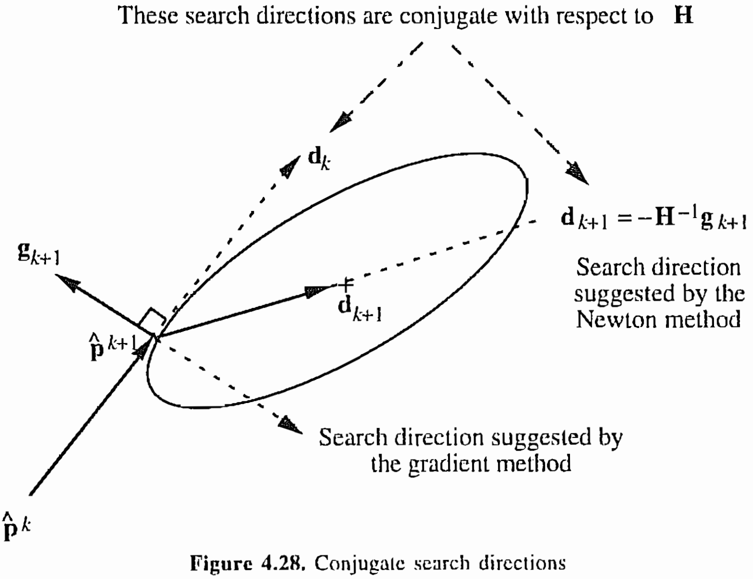

These search directions are conjugate with respect to H
$\mathbf{d}_k$
$\mathbf{d}_{k+1} = -\mathbf{H}^{-1}\mathbf{g}_{k+1}$
Search direction suggested by the Newton method
$\mathbf{g}_{k+1}$
$\mathbf{d}_{k+1}$
$\hat{\mathbf{p}}^{k+1}$
Search direction suggested by the gradient method
$\hat{\mathbf{p}}^{k}$

Figure 4.28. Conjugate search directions

The conjugate-gradient method aims at simple generation of n_{p} mutually conjugate search directions (see also Powell's method in Section 4.3.2.6). If the Hessian **H** were known, for any $\mathbf{d}_0$, for instance

$$\mathbf{d}_0 = -\mathbf{g}_0,$$

the algorithm

$$\lambda_k = -\frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{d}_k}{\mathbf{d}_k^{\mathrm{T}} \mathbf{H} \mathbf{d}_k},$$

$$\mathbf{g}_{k+1} = \mathbf{g}_k + \lambda_k \mathbf{H} \mathbf{d}_k,$$

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \frac{(\mathbf{g}_{k+1} - \mathbf{g}_k)^{\mathrm{T}}\mathbf{g}_{k+1}}{\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k}\mathbf{d}_k,$$

would ensure mutual conjugation, *i.e.*

$$\mathbf{d}_i^{\mathrm{T}}\mathbf{H}\mathbf{d}_k = 0 \;\; \forall\, i \neq k.$$

The proof of this result is by induction (Polak, 1971).

If the cost function is quadratic in $\mathbf{p}$, n_{p} one-dimensional minimizations along the $\mathbf{d}_i$'s ($i = 0, \ldots , n_{\mathrm{p}} - 1$) suffice to reach its minimum. Since $\mathbf{H}$ is unknown, the following result is used:

THEOREM

If $\hat{\mathbf{p}}^{k+1}$ is obtained by minimizing j along $\mathbf{d}_k$ from $\hat{\mathbf{p}}^k$, *i.e.* if

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k + \lambda^*\mathbf{d}_k,$$

with

$$\lambda^* = \arg\min_{\lambda} j(\hat{\mathbf{p}}^k + \lambda\mathbf{d}_k),$$

then $\mathbf{g}(\hat{\mathbf{p}}^{k+1})$ is identical to $\mathbf{g}_{k+1}$ as would be obtained by the formula requiring knowledge of $\mathbf{H}$. ◊

PROOF

From the second expression of the cost function, one gets

$$\mathbf{g}(\hat{\mathbf{p}}^k) = \mathbf{H}\hat{\mathbf{p}}^k + \mathbf{b},$$

$$\mathbf{g}(\hat{\mathbf{p}}^{k+1}) = \mathbf{H}\hat{\mathbf{p}}^{k+1} + \mathbf{b} = \mathbf{H}(\hat{\mathbf{p}}^k + \lambda^*\mathbf{d}_k) + \mathbf{b} = \mathbf{g}(\hat{\mathbf{p}}^k) + \lambda^*\mathbf{H}\mathbf{d}_k.$$

Now at the minimum of j along $\mathbf{d}_k$, the scalar product of $\mathbf{d}_k$ and the gradient is zero:

$$\mathbf{d}_k^{\mathrm{T}}\mathbf{g}(\hat{\mathbf{p}}^{k+1}) = 0.$$

Therefore

$$\lambda^* = -\frac{\mathbf{d}_k^{\mathrm{T}}\mathbf{g}(\hat{\mathbf{p}}^k)}{\mathbf{d}_k^{\mathrm{T}}\mathbf{H}\mathbf{d}_k},$$

i.e. the same expression as obtained for λ_k when $\mathbf{H}$ is known. ◊

Successive search directions can therefore be computed without knowing $\mathbf{H}$ or even trying to approximate its inverse as in the quasi-Newton methods. In summary, the *Polak-Ribière algorithm* (Polak, 1971) is given by

Step 0: Choose $\hat{\mathbf{p}}^0$ and evaluate $\mathbf{g}_0 = \mathbf{g}(\hat{\mathbf{p}}^0)$. If $\mathbf{g}_0 = \mathbf{0}$ stop.
Step 1: Set $k = 0$ and $\mathbf{d}_0 = -\mathbf{g}_0$.
Step 2: Compute

$$\lambda^* = \arg\min_{\lambda \geq 0} j(\hat{\mathbf{p}}^k + \lambda \mathbf{d}_k).$$

Step 3: Set $\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k + \lambda^* \mathbf{d}_k$.
Step 4: Evaluate $\mathbf{g}_{k+1} = \mathbf{g}(\hat{\mathbf{p}}^{k+1})$.
Step 5: If $\mathbf{g}_{k+1} = \mathbf{0}$ stop, else set

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \frac{(\mathbf{g}_{k+1} - \mathbf{g}_k)^{\mathrm{T}} \mathbf{g}_{k+1}}{\mathbf{g}_k^{\mathrm{T}} \mathbf{g}_k} \mathbf{d}_k,$$

increment k by one and go to Step 2.

Properties of conjugate-gradient algorithms. The first four properties are very close to those of quasi-Newton algorithms, in contrast to the others.

PCG1: Each iteration only requires the evaluation of the cost and its gradient. The computation is simple, with no set of linear equations in n_{p} unknowns to be solved.
PCG2: When the cost function is quadratic in $\mathbf{p}$, the algorithm converges to the minimum in n_{p} iterations.
PCG3: The initial search direction $\mathbf{d}_0$ may be the negative gradient direction. Since the cost is usually not quadratic in $\mathbf{p}$ far from the optimum, convergence to a local minimum is guaranteed only if the procedure is periodically restarted, say every n_{p} iterations, *e.g.* with a search from the current value of $\hat{\mathbf{p}}$ in the direction suggested by the gradient method.
PCG4: The method is more sensitive to errors in the evaluation of the gradient than the Gauss-Newton method.
PCG5: In contrast to quasi-Newton methods, conjugate-gradient algorithms do not update any approximation of the inverse of the Hessian. Each iteration is therefore even simpler. The simplification becomes more and more significant as the number of parameters increases. Conjugate-gradient methods have thus been successfully used to minimize cost functions depending on several thousands of parameters, *e.g.* in image processing. The price to be paid for this simplification is that the approximation of the inverse of the Hessian, which would be useful to characterize the uncertainty in the parameters, is no longer available.
PCG6: Conjugate-gradient methods require about n_{p} times as many one-dimensional minimizations as quasi-Newton methods for the same asymptotic behaviour (superlinear or quadratic convergence, depending on the hypotheses about the cost function) (Minoux, 1983). Note however that each iteration is much simpler.

EXAMPLE 4.10 (continued)

Figure 4.29 illustrates the minimization of Rosenbrock's function using the Polak-Ribière conjugate-gradient algorithm. The first step is taken in the direction opposite to the gradient. The line searches are performed with Brent's method using the derivative of the cost (Press *et al.*, 1986). After 289 evaluations of the cost function and 233 evaluations of its gradient, the value of the cost is 3.8×10^{-10}. ◊

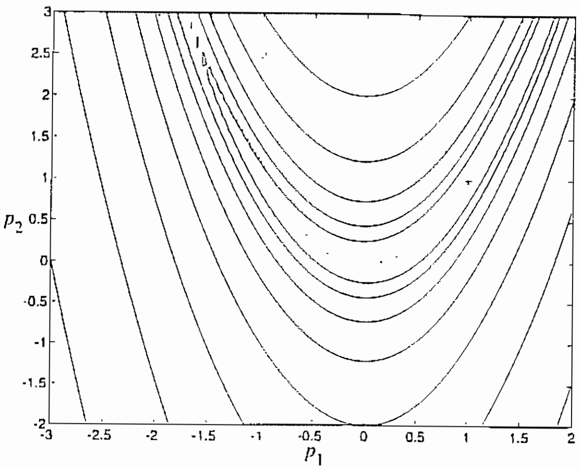


Figure 4.29. Behaviour of a conjugate-gradient method on Rosenbrock's function; the initial value of **p** is indicated by a circle, and the minimizer by a cross

EXAMPLE 4.11 (continued)

Consider again the cost function

$$j(\mathbf{p}) = (p_1 + p_2^2 - 5)^2 + (p_1 + p_2 - 2)^2 + (p_1 + p_2 - 4)^2$$

The trajectory of the same version of the Polak-Ribière algorithm as in Example 4.10 is illustrated by Figure 4.30. The minimum at $\hat{\mathbf{p}} = (1, 2)^{\mathrm{T}}$ is found with four significant digits after 55 evaluations of the cost, and 43 evaluations of its gradient. The one-dimensional searches were conducted much more carefully than in the quasi-Newton method (Brent's method with derivatives was used instead of Wolfe's algorithm), but performance was nevertheless poorer. ◊

As the quasi-Newton methods, conjugate-gradient methods are available in all serious libraries of scientific subroutines. It is therefore very simple to test whether they perform satisfactorily in the specific application at hand.

4.3.3.9 Choice of step size

After choosing a search direction $\mathbf{d}_k$ from the current point $\hat{\mathbf{p}}^k$ in parameter space, for example by Newton's method or the Gauss-Newton or quasi-Newton method, one must find a minimizer (or at least an acceptable value of the parameters) in this direction. Some of the methods presented in Section 4.3.2 do not use the local properties of the cost function. When the derivative of the cost is available, which is the case in the Newton, Gauss-Newton or quasi-Newton algorithms, it is more efficient to use it also during the one-dimensional searches.

One could of course search along $\mathbf{d}_k$ with the help of a one-dimensional gradient or Newton algorithm, but this approach turns out to yield rather slow convergence. As a matter of fact, what really matters is to ensure a *significant decrease* of the value of the cost in *as few iterations as possible*, so as not to dissipate effort in solving a purely local problem (Lemaréchal, 1989). When the cost has been evaluated at $\hat{\mathbf{p}}^k + \lambda_i \mathbf{d}_k$, one must therefore decide whether the step size λ_i is acceptable, and if not suggest another λ_{i+1}.

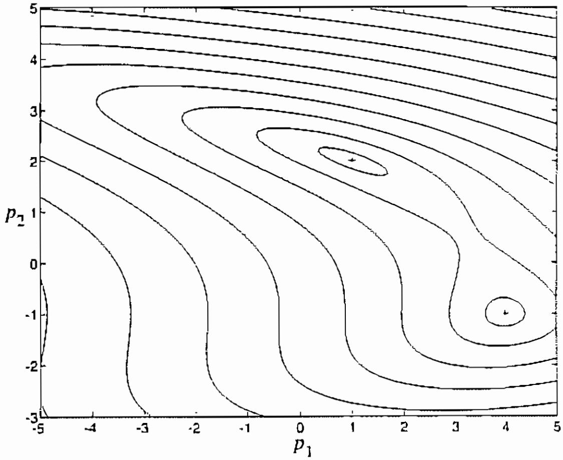


Figure 4.30. Behaviour of a conjugate-gradient algorithm on Example 4.11; the initial value of **p** is indicated by a circle, and the two global minimizers by crosses

REMARK 4.19

An initial step size λ_1 must also be chosen. For Newton-like algorithms using the Hessian (or an approximation of its inverse), the search direction is given by $\mathbf{d}_k = -\mathbf{H}^{-1}(\hat{\mathbf{p}}^k)\mathbf{g}(\hat{\mathbf{p}}^k)$, and $\lambda_1 = 1$ is a reasonable choice. For the conjugate-gradient algorithm, one could normalize $\|\mathbf{d}_k\|_2$ to one and choose λ_1 from prior knowledge of the feasible domain for the parameters. ◊

Wolfe's method is commonly used (Wolfe, 1969; Powell, 1976):

Step 0: Choose $\lambda_1 > 0$, α_1, α_2, with $0 < \alpha_1 < \alpha_2 < 1$ (*e.g.*, $\alpha_1 = 0.1$, $\alpha_2 = 0.5$). Set $\lambda_{\min} = \lambda_{\max} = 0$, $i = 1$.
Step 1: If $j(\hat{\mathbf{p}}^k + \lambda_i \mathbf{d}_k) > j(\hat{\mathbf{p}}^k) + \alpha_1 \lambda_i \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\mathbf{d}_k$ (λ_i is too large), set $\lambda_{\max} = \lambda_i$ and go to Step 4;
Step 2: If $\mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k + \lambda_i \mathbf{d}_k)\mathbf{d}_k \geq \alpha_2 \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\mathbf{d}_k$, set $\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k + \lambda_i \mathbf{d}_k$; else ($\lambda_i$ is too small), set $\lambda_{\min} = \lambda_i$.
Step 3: If $\lambda_{\max} = 0$, {set $\lambda_{i+1} = 2\lambda_i$, increment i by one and go to Step 1}.
Step 4: Set $\lambda_{i+1} = (\lambda_{\min} + \lambda_{\max})/2$, increment i by one and go to Step 1.

The principle of the algorithm is illustrated by Figure 4.31. Step 1 makes it possible to reject the values of $\lambda > b$, such that the decrease of the cost is insufficient. Step 2 allows rejection of the values of $\lambda < a$, for which $\hat{\mathbf{p}}^{k+1}$ could be too close to $\hat{\mathbf{p}}^k$ and thus $j(\hat{\mathbf{p}}^{k+1})$ too close to $j(\hat{\mathbf{p}}^k)$.

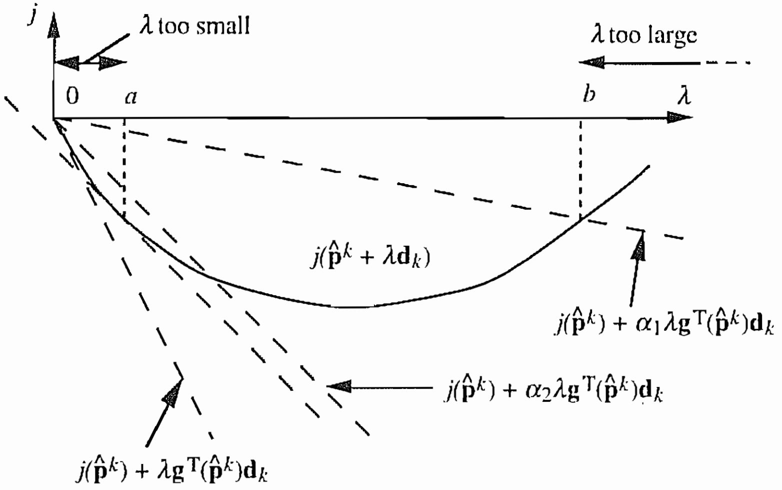

j
λ too small
λ too large
0
a
b
λ
$j(\hat{\mathbf{p}}^k + \lambda \mathbf{d}_k)$
$j(\hat{\mathbf{p}}^k) + \alpha_1 \lambda \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\mathbf{d}_k$
$j(\hat{\mathbf{p}}^k) + \alpha_2 \lambda \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\mathbf{d}_k$
$j(\hat{\mathbf{p}}^k) + \lambda \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\mathbf{d}_k$

Figure 4.31. Principle of Wolfe's procedure

This procedure terminates in a finite number of iterations for any cost function that can be bounded from below (such as a positive cost function).

REMARKS 4.20

— This one-dimensional search procedure is not necessarily very efficient for conjugate-gradient algorithms, for which careful minimization of the cost along $\mathbf{d}_k$ is advisable.

— The procedure extends without difficulty to the Levenberg-Marquardt method. Note, however, that the one-dimensional search on μ is no longer performed in a fixed direction in parameter space. ◊

4.3.4 Constrained optimization

As already mentioned in Remark 3.12, incorporating constraints on the feasible parameters into an estimation problem is not always to be recommended but may turn out to be necessary. Moreover, constraints are essential to many optimization problems, such as those encountered in experiment design (Chapter 6). Constrained optimization, however, is much more complicated than unconstrained optimization. Section 3.6 has shown how a constrained optimization problem can be transformed into an unconstrained one by modification of the cost function. Many other approaches may be considered, especially for inequality constraints (Polak, 1971; Luenberger, 1973; Gill and Murray, 1974; Minoux, 1983; Polyak, 1987; Hiriart-Urruty and Lemaréchal, 1993). In particular, the theoretical properties of the so-called *interior-point methods* have motivated a renewal of interest in convex programming (Nesterov and Nemirovskii, 1994; den Hertog, 1994). We shall only consider here *primal methods*, *i.e.* methods that search directly through the feasible domain for the optimal solution. They present the advantage of generating a sequence of feasible points, with decreasing values of the cost. The terminal point is thus always an improvement over the initial one, even when iterations are terminated too early. However, primal methods often exhibit slower convergence than primal-dual methods (Minoux, 1983), and the necessity of remaining in the feasible domain may produce computational difficulties (Luenberger, 1973; Minoux, 1983). Only the general ideas of the methods will be described, mainly in geometrical terms. Once again, we stress that many algorithms are readily available in scientific subroutine libraries, and can therefore be used at little or no programming cost.

Denote the set of all inequality constraints to be satisfied by $\mathbf{c}_i(\mathbf{p}) \leq \mathbf{0}$, to be understood componentwise, and let $\mathbb{P}$ be the corresponding prior feasible set for the parameters

$$\mathbb{P} = \{\mathbf{p} \mid \mathbf{c}_i(\mathbf{p}) \leq \mathbf{0}\}.$$

Whenever necessary, j and $\mathbf{c}_i$ will be assumed to be continuously differentiable. Some methods require $\mathbb{P}$ to have a nonempty interior $\mathbb{I}_\mathbb{P}$, the closure of which coincide with $\mathbb{P}$, because the optimizer will be searched for in $\mathbb{I}_\mathbb{P}$; see, *e.g.*, the method of centres. In some other methods, equality constraints can be taken into account simply by considering them as active inequality constraints. This is the case, *e.g.*, for the gradient-projection and feasible-directions methods.

The simplest constrained problem is when the cost and constraints are linear in the parameters. This corresponds to linear programming, considered in Section 4.3.4.1. The case where the cost is quadratic in the parameters corresponds to quadratic programming, treated in Section 4.3.4.2. When none of these approaches applies, but the shape of $\mathbb{P}$ is simple enough (*e.g.*, when $\mathbb{P}$ is an orthotope, which is quite frequent in parameter-estimation problems), direct extensions of unconstrained approaches such as those of Sections 4.3.4.3 to 4.3.4.5 can be considered. More general constrained optimization methods are finally presented in Sections 4.3.4.6 and 4.3.4.7.

4.3.4.1 Linear programming

Minimizing a linear cost function

$$j(\mathbf{p}) = \mathbf{c}^{\mathrm{T}}\mathbf{p}$$

under linear constraints

$$\mathbf{a}_i^{\mathrm{T}}\mathbf{p} \leq b_i \ (i = 1, \dots, m)$$

is a linear-programming problem, with countless practical applications. Famous methods, such as Dantzig's simplex (1963) and Karmarkar's algorithm (1984), are available but will not be described here. The latter belongs to the class of path-following methods; see the survey by Gonzaga (1992). We shall only present two simple approaches, particular cases of methods used in Section 5.4.1 to characterize parameter uncertainty in a bounded-error context. These approaches can be used in parameter estimation, where the number of variables is usually quite low compared to other types of applications. The first is recursive and based on polyhedra, the second non-recursive and based on ellipsoids.

The first method uses an exact description of a set defined by linear inequalities. The constraints $\mathbf{a}_i^{\mathrm{T}}\mathbf{p} \leq b_i$ $(i = 1, \dots, m)$ define a polyhedron $\mathbb{Q}_m$ in $\mathbb{R}^{n_\mathrm{p}}$, assumed to be compact (*i.e.* a polytope). This polytope can be built recursively, by introducing the constraints one by one, with the help of the method described in Section 5.4.1.3. The minima are either at the vertex associated with the smallest value of $\mathbf{c}^{\mathrm{T}}\mathbf{p}$ or in the convex hull of such vertices.

This recursive method is limited to low-dimensional problems. When recursive operation is not essential, the following approach may be preferred.

Step 0: Choose $\hat{\mathbf{p}}^0$, set $k = 0$.
Step 1: Compute $\hat{\mathbf{p}}^{k+1}$ such that

$$\hat{\mathbf{p}}^{k+1} \in \mathbb{Q}_m = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid \mathbf{a}_i^\mathrm{T}\mathbf{p} \leq b_i,\ i = 1, \ldots, m\} \quad \text{and} \quad \mathbf{c}^\mathrm{T}\hat{\mathbf{p}}^{k+1} < \mathbf{c}^\mathrm{T}\hat{\mathbf{p}}^k.$$

Step 2: Increment k by one and go to Step 1.

In Step 1, $\hat{\mathbf{p}}^{k+1}$ is obtained by an ellipsoidal algorithm, such as that to be presented in Section 4.3.5.3, with the subgradient $\tilde{\mathbf{g}}(\hat{\mathbf{p}}^k)$ replaced by $\mathbf{c}$ or by the vector $\mathbf{a}_i$ associated with the constraint most violated at $\hat{\mathbf{p}}^k$. Successive cuts in the ellipsoids are then *central*, *i.e.* the previous ellipsoid is cut through its centre. Shallower or deeper cuts may also be employed (Bland, Goldfarb and Todd, 1981; Grötschel, Lovasz and Schrijver, 1988). The ellipsoid obtained is given by

$$\mathbb{E}_{k+1} = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid (\mathbf{p} - \hat{\mathbf{p}}^{k+1})^\mathrm{T}\mathbf{M}_{k+1}^{-1}(\mathbf{p} - \hat{\mathbf{p}}^{k+1}) \leq 1\},$$

and contains the constrained minimizer. A lower bound for the value of the cost $\mathbf{c}^\mathrm{T}\mathbf{p}$ is therefore

$$\underline{j}_{k+1} = \max\{\underline{j}_k, \mathbf{c}^\mathrm{T}\hat{\mathbf{p}}^{k+1} - \sqrt{\mathbf{c}^\mathrm{T}\mathbf{M}_{k+1}\mathbf{c}}\ \} \leq \min_{\mathbf{p} \in \mathbb{Q}_m} \mathbf{c}^\mathrm{T}\mathbf{p},$$

with $\underline{j}_0$ chosen so that

$$\underline{j}_0 \leq \min_{\mathbf{p} \in \mathbb{Q}_m} \mathbf{c}^\mathrm{T}\mathbf{p}.$$

The optimal value of the cost is thus bracketed by

$$\underline{j}_{k+1} \leq \min_{\mathbf{p} \in \mathbb{Q}_m} \mathbf{c}^\mathrm{T}\mathbf{p} \leq \mathbf{c}^\mathrm{T}\hat{\mathbf{p}}^{k+1},$$

which can be used to stop the procedure when the precision reached is sufficient. Note that taking an extra constraint into account would require treating the problem again from scratch, unless $\hat{\mathbf{p}}^{k+1}$ satisfied this constraint.

This method has been the starting point for important theoretical work in optimization. It was used to prove that linear programs could be solved by algorithms with polynomial complexity (Khachiyan, 1979), whereas, in the worst case, the computational time of Dantzig's simplex algorithm grows exponentially with the dimension of the problem. Note, however, that Karmarkar's algorithm has less complexity than this ellipsoidal algorithm.

Application to L_∞ estimation. The L_∞ (or minimax) estimator has been introduced in Section 3.3.1, Example 3.6. It is given by

$$\hat{\mathbf{p}}_\mathrm{mm} = \arg \min_{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}}} \max_{1 \leq i \leq n_\mathrm{t}} |y(t_i) - y_\mathrm{m}(t_i, \mathbf{p})|,$$

for an output error, but prediction errors could be considered instead. Consider an LP model structure defined by

$$y_{\mathrm{m}}(t_i, \mathbf{p}) = \mathbf{r}_i^{\mathrm{T}}\mathbf{p},\ i = 1, \ldots, n_{\mathrm{t}}.$$

Let

$$\hat{a}_{\mathrm{min}} = \max_{1 \leq i \leq n_{\mathrm{t}}} |y(t_i) - y_{\mathrm{m}}(t_i, \hat{\mathbf{p}}_{\mathrm{mm}})|,$$

and consider the set of all parameter vectors **p** that satisfy

$$|y(t_i) - \mathbf{r}_i^{\mathrm{T}}\mathbf{p}| \leq a,\ i = 1, \ldots, n_{\mathrm{t}}.$$

If $a \geq \hat{a}_{\mathrm{min}}$, this set is non-empty, and corresponds to the *posterior feasible set* $\mathbb{P}_{n_{\mathrm{t}}}$ for errors bounded by a (Section 5.4). Assume that the $\mathbf{r}_i$'s span $\mathbb{R}^{n_{\mathrm{p}}}$, an identifiability condition for the parameters. $\mathbb{P}_{n_{\mathrm{t}}}$ is then a polytope (bounded polyhedron) of $\mathbb{R}^{n_{\mathrm{p}}}$. Consider the extended parameter vector $\mathbf{p}_{\mathrm{e}} = (\mathbf{p}^{\mathrm{T}}, a)^{\mathrm{T}}$ of $\mathbb{R}^{n_{\mathrm{p}}+1}$. If a (possibly very large) upper bound a_{max} is available for a, the set of all $\mathbf{p}_{\mathrm{e}}$ that satisfy

$$|y(t_i) - \mathbf{r}_i^{\mathrm{T}}\mathbf{p}| \leq a,\ i = 1, \ldots, n_{\mathrm{t}} \quad \text{and} \quad a \leq a_{\mathrm{max}}$$

is a polytope of $\mathbb{R}^{n_{\mathrm{p}}+1}$, which can be built recursively as the data are collected. The estimate $\hat{\mathbf{p}}_{\mathrm{mm}}$ corresponds either to the vertex of this polytope at which $a = \hat{a}_{\mathrm{min}}$, the smallest value over all vertices, or to the convex hull of all vertices associated with $\hat{a}_{\mathrm{min}}$. It can therefore be obtained recursively (Walter and Piet-Lahanier, 1991).

Since minimax or L_∞ estimation corresponds to minimizing a under the previous constraints, the ellipsoidal approach presented above may replace the polyhedral one when recursive operation is not essential.

EXAMPLE 4.16

Consider the AR structure described by

$$y(k) = -0.4y(k-1) - 0.85y(k-2) + \varepsilon(k),$$

with $y(1) = \varepsilon(1)$ and $y(2) = \varepsilon(2)$. Figure 4.32 presents the evolution of $\hat{a}_{\mathrm{min}}$ with k when $\{\varepsilon(k)\}$ is a sequence of independent random variables, distributed either $\mathcal{N}(0, 1)$ or $\mathcal{U}(-1, 1)$. When the errors are indeed bounded, $\hat{a}_{\mathrm{min}}$ is seen to approach the actual bound quickly. ◊

4.3.4.2 Quadratic programming

Quadratic programs are involved, for instance, in least-squares estimation under linear constraints (Section 4.1.3) or in the implementation of the constrained Newton method to be presented in Section 4.3.4.5.

Assume that the cost to be minimized is

$$j(\mathbf{p}) = \frac{1}{2}\,\mathbf{p}^{\mathrm{T}}\mathbf{C}\mathbf{p} - \mathbf{d}^{\mathrm{T}}\mathbf{p},$$

with $\mathbf{C}$ positive definite (so that j is convex), under the same constraints as in the linear-programming case, *i.e.* $\mathbf{Ap} \leq \mathbf{b}$, which define a feasible set $\mathbb{Q}_m$. Let $\hat{\mathbf{p}}^k$ be a feasible parameter vector, and $\mathbb{I}_k$ be the set of all active constraints at $\hat{\mathbf{p}}^k$:

$$\mathbb{I}_k = \{ i \mid \mathbf{a}_i^{\mathrm{T}} \hat{\mathbf{p}}^k = b_i \}.$$

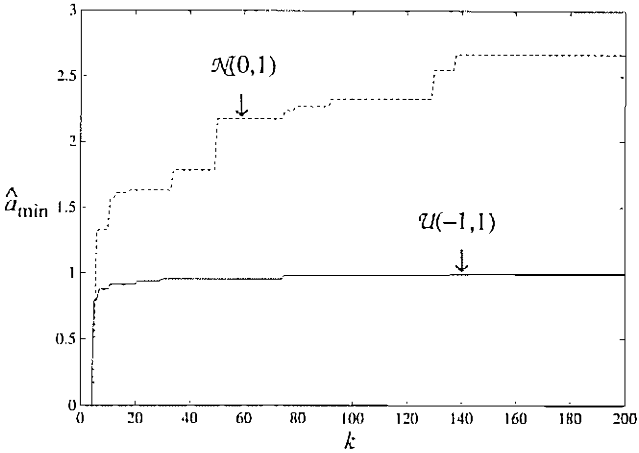


Figure 4.32. Evolution of $\hat{a}_{\min}$ with the number of data points

A classical algorithm searches for $\hat{\mathbf{p}}^{k+}$ that minimizes $j(\mathbf{p})$ under all active constraints considered as equality constraints. If $\mathbf{A}_k$ and $\mathbf{b}_k$ consist of those rows of $\mathbf{A}$ and $\mathbf{b}$ associated with active constraints, the Lagrangian of this problem can be written as:

$$\mathcal{L}(\mathbf{p}, \mathbf{z}) = \frac{1}{2}\, \mathbf{p}^{\mathrm{T}}\mathbf{C}\mathbf{p} - \mathbf{d}^{\mathrm{T}}\mathbf{p} + \mathbf{z}^{\mathrm{T}}(\mathbf{A}_k - \mathbf{b}_k).$$

Stationarity with respect to $\mathbf{p}$ and $\mathbf{z}$ of $\mathcal{L}(\mathbf{p}, \mathbf{z})$ at the optimum yields a set of linear equations to be satisfied by $\hat{\mathbf{p}}^{k+}$ and $\hat{\mathbf{z}}$:

$$\begin{bmatrix} \mathbf{C} & \mathbf{A}_k^{\mathrm{T}} \\ \mathbf{A}_k & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}}^{k+} \\ \hat{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ \mathbf{b}_k \end{bmatrix}.$$

If $\mathbf{A}\hat{\mathbf{p}}^{k+} \leq \mathbf{b}$ and $\hat{\mathbf{z}} \geq 0$, $\hat{\mathbf{p}}^{k+}$ is a solution of the quadratic programming problem. If $\hat{\mathbf{p}}^{k+}$ violates some constraints, a search is performed on the line segment joining $\hat{\mathbf{p}}^k$ to $\hat{\mathbf{p}}^{k+}$ to get a new feasible point $\hat{\mathbf{p}}^{k+1}$. Any new active constraint is then incorporated in $\mathbb{I}_{k+1}$. If some components of $\hat{\mathbf{z}}$ turn out to be negative, the associated constraints are dropped from $\mathbb{I}_k$ before restarting from $\hat{\mathbf{p}}^k$. The resulting algorithm finds a solution in a finite number of steps (Polyak, 1987):

Step 0: Choose a feasible $\hat{\mathbf{p}}^0$ ($\mathbf{A}\hat{\mathbf{p}}^0 \leq \mathbf{b}$), set $k = 0$.
Step 1: Find the set $\mathbb{I}_k$ of all active constraints at $\hat{\mathbf{p}}^k$
Step 2: Compute $\hat{\mathbf{p}}^{k+}$ and $\hat{\mathbf{z}}$. The nth component of $\hat{\mathbf{z}}$ is associated with an element i of $\mathbb{I}_k$, which will be denoted by $n(i)$. If $\mathbf{A}\hat{\mathbf{p}}^{k+} \leq \mathbf{b}$, go to Step 4.
Step 3: Set

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k + \lambda_k(\hat{\mathbf{p}}^{k+} - \hat{\mathbf{p}}^k),$$

with

$$\lambda_k = \max\{\lambda \mid \hat{\mathbf{p}}^k + \lambda(\hat{\mathbf{p}}^{k+} - \hat{\mathbf{p}}^k) \in \mathbb{Q}_m\}.$$

Set $\mathbb{I}_{k+1} = \mathbb{I}_k \cup \{n \mid n \notin \mathbb{I}_k, \mathbf{a}_n^{\mathrm{T}}\hat{\mathbf{p}}^{k+1} = b_n\}$, increment k by one and go to Step 2.
Step 4: If $\hat{z}_{n(i)} \geq 0$ for all i in $\mathbb{I}_k$ then $\hat{\mathbf{p}} = \hat{\mathbf{p}}^{k+}$, stop. Else set $\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k$, $\mathbb{I}_{k+1} = \{i \in \mathbb{I}_k \mid \hat{z}_{n(i)} > 0\}$, increment k by one and go to Step 2.

More general convex-programming algorithms can, of course, also be used for convex quadratic programming; see Section 4.3.4.6 and the references on interior-point methods therein.

4.3.4.3 Constrained gradient

The gradient algorithm is described by

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \lambda_k \mathbf{g}(\hat{\mathbf{p}}^k),$$

with λ_k chosen so as to minimize $j(\hat{\mathbf{p}}^{k+1})$. Under constraints, $-\mathbf{g}(\hat{\mathbf{p}}^k)$ might point in an unfeasible direction that would entail leaving $\mathbb{P}$. It then seems natural to replace $-\mathbf{g}(\hat{\mathbf{p}}^k)$ in the previous equation by the direction $\mathbf{d}_k$ given by

$$\mathbf{d}_k = \hat{\mathbf{p}}^{k+} - \hat{\mathbf{p}}^k,$$

with $\hat{\mathbf{p}}^{k+}$ obtained by minimizing the *linear* cost function $\mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)(\mathbf{p} - \hat{\mathbf{p}}^k)$, *i.e.*

$$\hat{\mathbf{p}}^{k+} = \arg \min_{\mathbf{p} \in \mathbb{P}} \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)(\mathbf{p} - \hat{\mathbf{p}}^k).$$

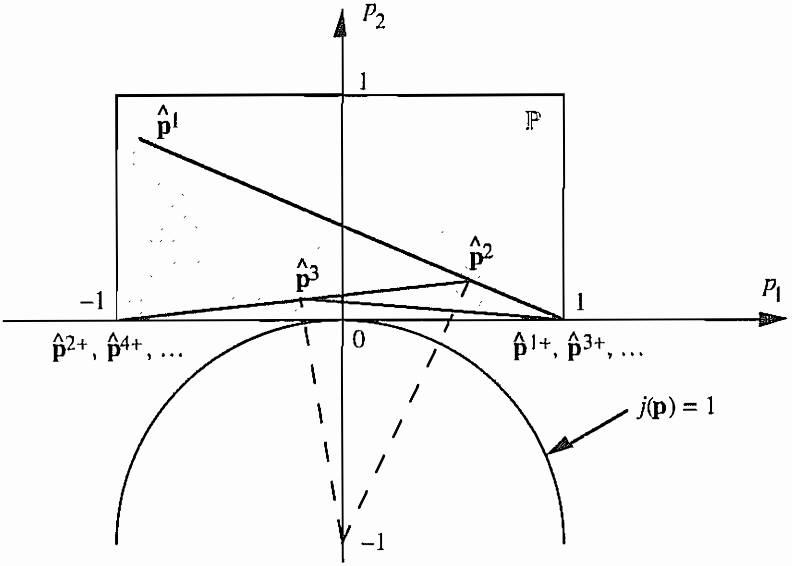
This method is, however, not recommended, for it may converge very slowly, as illustrated by the following simple example (Polyak, 1987).

EXAMPLE 4.17

Consider the cost function

$$j(\mathbf{p}) = p_1^2 + (1 + p_2)^2, \text{ with } \mathbf{p} = (p_1, p_2)^{\mathrm{T}}, \text{ and } \mathbb{P} = \{\mathbf{p} \mid |p_1| \leq 1, 0 \leq p_2 \leq 1\}.$$

The vectors $\hat{\mathbf{p}}^{k+}$ alternate between vertices $(-1, 0)^{\mathrm{T}}$ and $(1, 0)^{\mathrm{T}}$ of the box $\mathbb{P}$, whereas the vectors $\hat{\mathbf{p}}^k$, located inside $\mathbb{P}$, converge very slowly to $(0, 0)^{\mathrm{T}}$ (Figure 4.33). The same type of behaviour takes place whenever $\mathbb{P}$ is a polytope with the minimum of j on one of its faces. ◊



p_2
1
$\mathbb{P}$
$\hat{\mathbf{p}}^1$
$\hat{\mathbf{p}}^2$
$\hat{\mathbf{p}}^3$
−1
1
p_1
$\hat{\mathbf{p}}^{2+}, \hat{\mathbf{p}}^{4+}, \ldots$
0
$\hat{\mathbf{p}}^{1+}, \hat{\mathbf{p}}^{3+}, \ldots$
$j(\mathbf{p}) = 1$
−1

Figure 4.33. Very slow convergence of the constrained-gradient method

4.3.4.4 Gradient-projection method

In this method (Rosen, 1960, 1961), the new parameter vector suggested by the gradient method, *i.e.* $\hat{\mathbf{p}}^k - \lambda\mathbf{g}(\hat{\mathbf{p}}^k)$, is projected onto $\mathbb{P}$, according to

$$\hat{\mathbf{p}}^{k+1}(\lambda) = \mathbf{O}^{\perp}[\hat{\mathbf{p}}^k - \lambda\mathbf{g}(\hat{\mathbf{p}}^k)] = \arg \min_{\mathbf{p}\in\mathbb{P}} \| \mathbf{p} - [\hat{\mathbf{p}}^k - \lambda\mathbf{g}(\hat{\mathbf{p}}^k)] \|_2^2,$$

where $\mathbf{O}^{\perp}$ is the orthogonal projector onto $\mathbb{P}$. The cost $j[\hat{\mathbf{p}}^{k+1}(\lambda)]$ is then minimized with respect to the scalar λ. This policy is illustrated by Figure 4.34.

Implementing the method involves solving the problem of projecting onto $\mathbb{P}$. Note that the optimal step size λ_k obtained by minimizing $j[\hat{\mathbf{p}}^{k+1}(\lambda)]$ may be infinite. In Figure 4.34, for example, $j(\hat{\mathbf{p}}^2)$ decreases as the point A moves away from $\hat{\mathbf{p}}^1$ in the direction $-\mathbf{g}(\hat{\mathbf{p}}^1)$. Choosing λ_1 optimally would lead to taking $\hat{\mathbf{p}}^2$ as close as possible to $\hat{\mathbf{p}}^\infty$, *i.e.* to building $\hat{\mathbf{p}}^2$ by projection along a direction almost parallel to $\mathbf{g}(\hat{\mathbf{p}}^1)$, which would mean taking λ_1 extremely large. A suboptimal step size must then be used, as in Figure 4.34.

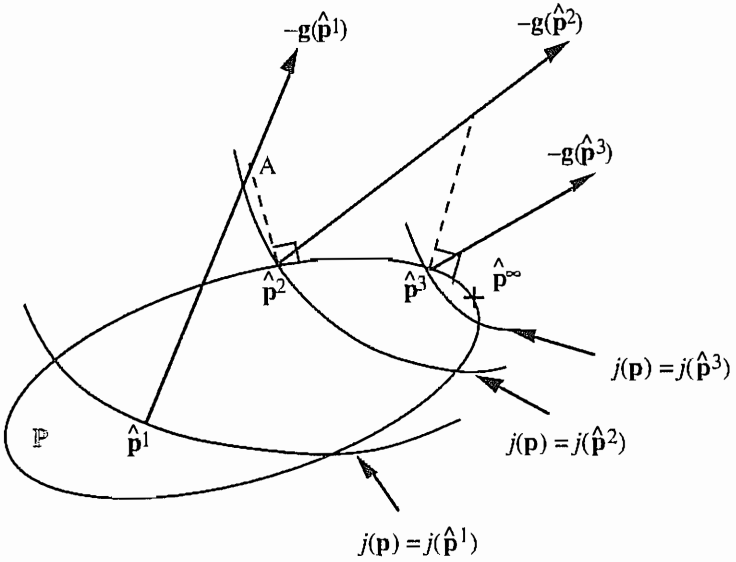
This method is much more efficient than the constrained-gradient method, as illustrated on Example 4.17 by Figure 4.35. Convergence is now obtained in one iteration (compare with Figure 4.33).

In practice, the method is easy to use only when $\mathbb{P}$ has a simple geometrical shape. If, for example, $\mathbb{P}$ is defined by the constraints

$$a_i \leq p_i \leq b_i,\ i \in \mathbb{I},\ \text{card}\ \mathbb{I} \leq n_{\mathrm{p}},$$

the components of $\hat{\mathbf{p}}^{k+1}(\lambda)$ can be chosen as

$$\begin{aligned} \hat{p}_i^{k+1}(\lambda) &= \hat{p}_i^k - \lambda g_i(\hat{\mathbf{p}}^k), \text{ if } a_i \leq \hat{p}_i^k - \lambda g_i(\hat{\mathbf{p}}^k) \leq b_i \text{ or } i \notin \mathbb{I} \\ \hat{p}_i^{k+1}(\lambda) &= b_i, \text{ if } \hat{p}_i^k - \lambda g_i(\hat{\mathbf{p}}^k) > b_i, \\ \hat{p}_i^{k+1}(\lambda) &= a_i, \text{ if } \hat{p}_i^k - \lambda g_i(\hat{\mathbf{p}}^k) < a_i. \end{aligned}$$



$-\mathbf{g}(\hat{\mathbf{p}}^1)$
$-\mathbf{g}(\hat{\mathbf{p}}^2)$
$-\mathbf{g}(\hat{\mathbf{p}}^3)$
A
$\hat{\mathbf{p}}^2$
$\hat{\mathbf{p}}^3$
$\hat{\mathbf{p}}^\infty$
$j(\mathbf{p}) = j(\hat{\mathbf{p}}^3)$
$j(\mathbf{p}) = j(\hat{\mathbf{p}}^2)$
$\mathbb{P}$
$\hat{\mathbf{p}}^1$
$j(\mathbf{p}) = j(\hat{\mathbf{p}}^1)$

Figure 4.34. Gradient-projection method (with suboptimal step size λ_k)

The search direction is thus modified only when constraints are active. This corresponds to the following algorithm, where the active or inactive character of the constraints is used to choose the direction $\mathbf{d}_k$.

Step 0: Choose $\hat{\mathbf{p}}^0 \in \mathbb{P}$, set $k = 0$.

Step 1: Compute $\mathbf{g}(\hat{\mathbf{p}}^k)$. Find the set $\mathbb{I}_k$ of all constraints which would be violated after any displacement along $-\mathbf{g}(\hat{\mathbf{p}}^k)$, *i.e.*

$$\mathbb{I}_\mathrm{k} = \{i \mid (\hat{p}_i^k = a_i \text{ and } g_i(\hat{\mathbf{p}}^k) > 0) \text{ or } (\hat{p}_i^k = b_i \text{ and } g_i(\hat{\mathbf{p}}^k) < 0)\}.$$

Step 2: Take the constraints in $\mathbb{I}_k$ into account to define the search direction $\mathbf{d}_k$

$$\text{if } i \in \mathbb{I}_k, \text{ then } d_{k_i} = 0, \text{ else } d_{k_i} = -g_i(\hat{\mathbf{p}}^k).$$

If $\|\mathbf{d}_k\| = 0$, stop.

Step 3: Compute $\hat{\mathbf{p}}^{k+1}(\lambda_k) = \hat{\mathbf{p}}^k + \lambda_k \mathbf{d}_k$, with

$$\lambda_k = \arg \min_{\lambda \mid \hat{\mathbf{p}}^{k+1}(\lambda) \in \mathbb{P}} j[\hat{\mathbf{p}}^{k+1}(\lambda)].$$

Increment k by one and go to Step 1.

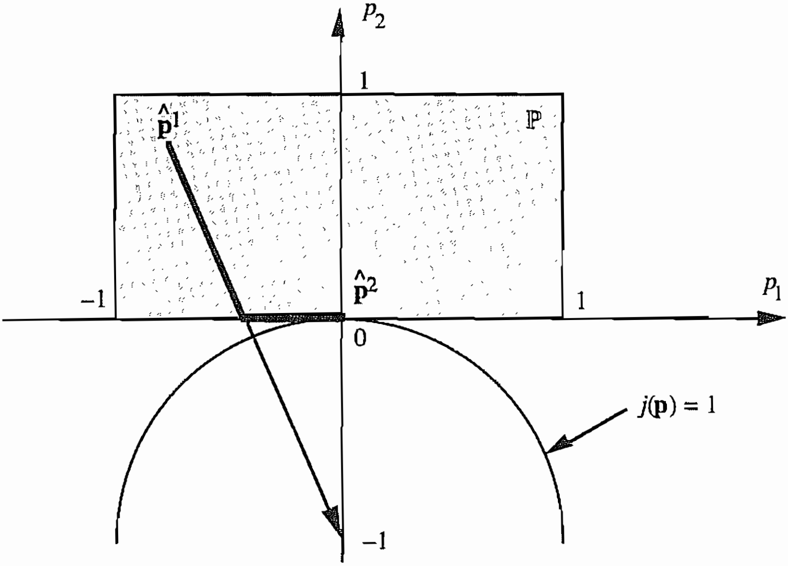


Figure 4.35. Behaviour of the gradient-projection algorithm

The stopping rule in Step 2 should be augmented by a test on the decrease of $j(\hat{\mathbf{p}}^k)$ between two consecutive iterations. See also Section 4.3.7 for stopping rules taking computer accuracy into account. This algorithm proceeds from the method of feasible directions, to be presented in Section 4.3.4.7 (without requiring solution of a linear program at each step). It can be generalized (Polak, 1971; Luenberger, 1973; Minoux, 1983), to constraints affine in **p** as well as to nonlinear constraints, although these raise specific difficulties. Surprisingly, neither a proof of convergence towards a local optimum nor a counterexample seems to be available (Luenberger, 1973; Minoux, 1983).

Note, finally, that a projected Newton method *may not converge*, as illustrated by Figure 4.36. This will hold true for any method based upon the recurrence

$$\hat{\mathbf{p}}^{k+1}(\lambda) = \mathbf{O}^{\perp}[\hat{\mathbf{p}}^k - \lambda \mathbf{H}\mathbf{g}(\hat{\mathbf{p}}^k)],$$

with $\mathbf{H} \neq \mathbf{I}_{n_\mathrm{p}}$ positive-definite (Polyak, 1987).

4.3.4.5 Constrained Newton and quasi-Newton

These approaches are based on quadratic approximations of the cost, and compute

$$\hat{\mathbf{p}}^{k+1} = \arg\min_{\mathbf{p}\in\mathbb{P}} \left[(\mathbf{p}-\hat{\mathbf{p}}^k)^{\mathrm{T}}\mathbf{g}(\hat{\mathbf{p}}^k) + \frac{1}{2}(\mathbf{p}-\hat{\mathbf{p}}^k)^{\mathrm{T}}\mathbf{H}(\mathbf{p}-\hat{\mathbf{p}}^k)\right].$$

The constrained Newton method is obtained when $\mathbf{H}$ is the Hessian at $\hat{\mathbf{p}}^k$. A Gauss-Newton approximation of the Hessian can also be considered, or an approximation of its inverse built recursively as in the quasi-Newton methods of Section 4.3.3.6.

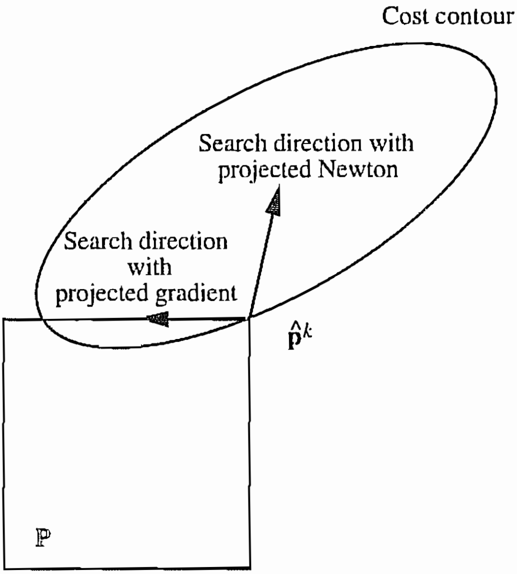

Cost contour
Search direction with projected Newton
Search direction with projected gradient
$\hat{\mathbf{p}}^k$
$\mathbb{P}$

Figure 4.36. Failure of the projected Newton method

REMARK 4.21

When $\mathbf{H} = (1/\lambda_k)\mathbf{I}_{n_\mathrm{p}}$, $\hat{\mathbf{p}}^{k+1}$ is given by

$$\begin{aligned}\hat{\mathbf{p}}^{k+1} &= \arg\min_{\mathbf{p}\in\mathbb{P}} \left[2\lambda_k(\mathbf{p}-\hat{\mathbf{p}}^k)^{\mathrm{T}}\mathbf{g}(\hat{\mathbf{p}}^k) + (\mathbf{p}-\hat{\mathbf{p}}^k)^{\mathrm{T}}(\mathbf{p}-\hat{\mathbf{p}}^k)\right]\\ &= \arg\min_{\mathbf{p}\in\mathbb{P}} \|\mathbf{p}-\hat{\mathbf{p}}^k + \lambda_k\mathbf{g}(\hat{\mathbf{p}}^k)\|_2^2,\end{aligned}$$

so the method boils down to gradient projection, provided that λ_k is chosen to minimize $j(\hat{\mathbf{p}}^{k+1})$. ◊

A possible variant computes

$$\hat{\mathbf{p}}^{k+} = \arg\min_{\mathbf{p}\in\mathbb{P}} \left[(\mathbf{p}-\hat{\mathbf{p}}^k)^{\mathrm{T}}\mathbf{g}(\hat{\mathbf{p}}^k) + \frac{1}{2}(\mathbf{p}-\hat{\mathbf{p}}^k)^{\mathrm{T}}\mathbf{H}(\mathbf{p}-\hat{\mathbf{p}}^k)\right],$$

and searches for $\hat{\mathbf{p}}^{k+1}$ along $\hat{\mathbf{p}}^{k+} - \hat{\mathbf{p}}^{k}$. Note that choosing $\mathbf{H} = \mathbf{0}$ then leads to the constrained-gradient method, found in Section 4.3.4.3 to be unsatisfactory.

Convergence is guaranteed only if $\mathbb{P}$ is convex. The search for $\hat{\mathbf{p}}^{k+}$ may not be trivial; see (Lawson and Hanson, 1974) for the solution of constrained least-squares problems. When the constraints are linear inequalities, the problem is quadratic programming, for which finite algorithms are available (Section 4.3.4.2). The constrained Newton method is then called *sequential quadratic programming*.

A constrained variant of the conjugate-gradient method is also available (Polyak, 1987) for problems where the constraints are

$$a_i \leq p_i \leq b_i,\ i \in \mathbb{I},\ \text{card}\ \mathbb{I} \leq \dim \mathbf{p}.$$

More general methods, which can be used when $\mathbb{P}$ is still convex but may have a more complicated shape, will now be presented.

4.3.4.6 Method of centres

Assume that there exists a vector $\hat{\mathbf{p}}^0$ such that the subset of $\mathbb{P}$ defined as

$$\mathbb{C}(\hat{\mathbf{p}}^0) = \{\mathbf{p} \mid j(\mathbf{p}) \leq j(\hat{\mathbf{p}}^0),\ \mathbf{c}_{\mathrm{i}}(\mathbf{p}) \leq \mathbf{0}\}$$

is compact with a nonempty interior. Assume moreover that $\mathbf{c}_{\mathrm{i}}(\mathbf{p}) < \mathbf{0}$ for any $\mathbf{p}$ inside $\mathbb{C}$. A point $\hat{\mathbf{p}}^1$ is generated at the Chebyshev centre of $\mathbb{C}(\hat{\mathbf{p}}^0)$, *i.e.* at a maximal distance from its boundary. A possible distance function is

$$d(\mathbf{p}', \mathbf{p}) = -\max\ \{j(\mathbf{p}') - j(\mathbf{p}),\ c_{\mathrm{i}_n}(\mathbf{p}'),\ n = 1, \ldots, \dim \mathbf{c}_{\mathrm{i}}\}.$$

The principle of the method is then:

Step 1: Choose $\hat{\mathbf{p}}^0$ such that $\mathbb{C}(\hat{\mathbf{p}}^0)$ is compact, with a nonempty interior. Set $k = 0$.
Step 2: Compute

$$\hat{\mathbf{p}}^{k+} = \arg \max_{\mathbf{p} \in \mathbb{R}^{n_{\mathrm{p}}}} d(\mathbf{p}, \hat{\mathbf{p}}^k).$$

Step 3: If $d(\hat{\mathbf{p}}^{k+}, \hat{\mathbf{p}}^k) = 0$ stop; else set $\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^{k+}$, increment k by one and go to Step 2.

In practice, the computation of $\hat{\mathbf{p}}^{k+}$ is performed by one-dimensional maximization of $d(\mathbf{p}, \hat{\mathbf{p}}^k)$ in a privileged direction $\mathbf{d}_k$. This direction must be such that for sufficiently small λ

$$j(\hat{\mathbf{p}}^k + \lambda \mathbf{d}_k) < j(\hat{\mathbf{p}}^k) \quad \text{and} \quad c_{\mathrm{i}_n}(\hat{\mathbf{p}}^k + \lambda \mathbf{d}_k) \leq \mathbf{0} \text{ if } c_{\mathrm{i}_n}(\hat{\mathbf{p}}^k) = 0.$$

A suitable direction (Polak, 1971) is given by the argument of the minimum over $\mathbf{d}$ of the cost

$$j'(\mathbf{d}) = \max\{\mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\mathbf{d},\ c_{\mathrm{i}_n}(\hat{\mathbf{p}}^k) + \mathbf{g}^{\mathrm{T}}_{c_{\mathrm{i}_n}}(\hat{\mathbf{p}}^k)\mathbf{d},\ n = 1, \ldots, \dim \mathbf{c}_{\mathrm{i}}\},$$

under the constraints $|d_i| \leq 1$ ($i = 1, \ldots, \dim \mathbf{p}$), with

$$\mathbf{g}_{c_{\mathrm{i}_n}}(\hat{\mathbf{p}}^k) = \frac{\partial c_{\mathrm{i}_n}(\mathbf{p})}{\partial \mathbf{p}}|_{\hat{\mathbf{p}}^k}.$$

It is obtained (Topkis and Veinnot, 1967; Minoux, 1983) by solving the following linear-programming problem (Section 4.3.4.1):

minimize α with respect to **d** under the constraints

$$\mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^k)\mathbf{d} \leq \alpha,$$

$$c_{\mathrm{i}_n}(\hat{\mathbf{p}}^k) + \mathbf{g}^{\mathrm{T}}_{c_{\mathrm{i}_n}}(\hat{\mathbf{p}}^k)\mathbf{d} \leq \alpha, \quad n = 1, \ldots, \dim \mathbf{c},$$

$$|d_i| \leq 1, \quad i = 1, \ldots, \dim \mathbf{p}.$$

When the optimal value of α is negative, the cost decreases in the direction **d**, which is therefore feasible in the sense that a small displacement along **d** keeps **p** feasible. When j and $\mathbf{c}_{\mathrm{i}}$ are convex, $d(., \hat{\mathbf{p}}^k)$ is convex too (but not differentiable everywhere), so minimization along $\mathbf{d}_k$ can be performed with the techniques proposed in Section 4.3.2.

When the constraints $\mathbf{c}_{\mathrm{i}}$ are linear and j is convex, $\mathbb{C}(\hat{\mathbf{p}}^k)$ is contained in the polytope

$$\mathbb{P}_k = \{\mathbf{p} \mid \mathbf{c}_{\mathrm{i}}(\mathbf{p}) \leq \mathbf{0}, \mathbf{g}^{\mathrm{T}}(\hat{\mathbf{p}}^n)(\mathbf{p} - \hat{\mathbf{p}}^n) \leq 0, n = 1, \ldots, k\}$$

One can then show (Levin, 1965) that setting $\hat{\mathbf{p}}^{k+1}$ at the centre of gravity of $\mathbb{P}_k$ (which is necessarily feasible) ensures

$$\mathrm{vol}(\mathbb{P}_{k+1}) \leq \left[1 - (1 - \frac{1}{n_{\mathrm{p}} + 1})^{n_{\mathrm{p}}}\right] \mathrm{vol}(\mathbb{P}_k), \text{ with } n_{\mathrm{p}} = \dim \mathbf{p},$$

and thus,

$$\mathrm{vol}(\mathbb{P}_{k+1}) \leq \left(1 - \frac{1}{e}\right) \mathrm{vol}(\mathbb{P}_k) \approx 0.632 \ \mathrm{vol}(\mathbb{P}_k).$$

This provides the fastest convergence, in terms of number of iterations, among all procedures using gradients only. However, computing the centre of gravity of a polytope is a heavy task, so the method is of no practical interest as it stands. This motivates taking as $\hat{\mathbf{p}}^{k+1}$ the centre of the maximal-volume ellipsoid inscribed in $\mathbb{P}_k$ (Tarasov, Khachiyan and Èrlikh, 1988), the determination of which is of polynomial complexity. See also (Khachiyan and Todd, 1993). A similar method can be used on cost functions not differentiable everywhere, simply by replacing **g** by a subgradient $\tilde{\mathbf{g}}$ (Section 4.3.5.1). When the constraints $\mathbf{c}_{\mathrm{i}}$ are nonlinear but convex, they can be linearized. If $\hat{\mathbf{p}}^{k+1}$ as determined from $\mathbb{P}_{k+1}$ does not satisfy a constraint $c_{\mathrm{i}_n}(\mathbf{p}) \leq 0$, the set

$$\mathbb{P}_{k+1} \cap \{\mathbf{p} \mid \mathbf{g}^{\mathrm{T}}_{c_{\mathrm{i}_n}}(\hat{\mathbf{p}}^{k+1})\,(\mathbf{p} - \hat{\mathbf{p}}^{k+1}) \leq 0\}$$

is substituted for $\mathbb{P}_{k+1}$ and a new vector $\hat{\mathbf{p}}^{k+1}$ is determined.

The attractive complexity properties of such interior-point methods have stimulated intense activity (den Hertog, 1994; Nesterov and Nemirovskii, 1994). Other methods can be considered for generating points interior to $\mathbb{P}_k$, such as the ellipsoidal method of Section 4.3.5.3.

4.3.4.7 Method of feasible directions

Although developed independently (Zoutendijk, 1960), this method is very close to the previous one. The vector $\hat{\mathbf{p}}^{k+}$ at Step 2 is now defined by

$$\hat{\mathbf{p}}^{k+} = \hat{\mathbf{p}}^{k} + \lambda_k \mathbf{d}_k.$$

The search direction $\mathbf{d}_k$ should satisfy $|d_{k_i}| \leq 1$, $i = 1, \dots, n_\mathrm{p}$, to normalize it with linear constraints. It should be feasible, *i.e.* such that

$$\mathbf{g}^\mathrm{T}(\hat{\mathbf{p}}^k)\mathbf{d}_k \leq 0, \quad \text{and} \quad \mathbf{g}^\mathrm{T}_{c_{\mathrm{i}_n}}(\hat{\mathbf{p}}^k)\mathbf{d}_k \leq 0 \text{ for all active constraints } c_{\mathrm{i}_n}.$$

Thus, the cost can be decreased with $c_{\mathrm{i}_n}(\mathbf{p}) \leq 0$ for small enough displacements along $\mathbf{d}_k$.

This suggests that $\mathbf{d}_k$ should be chosen as the solution of the following linear-programming problem:

$$\mathbf{d}_k = \arg \min_{\mathbf{d}} \mathbf{g}^\mathrm{T}(\hat{\mathbf{p}}^k)\mathbf{d},$$

with the constraints

$$\mathbf{g}^\mathrm{T}_{c_{\mathrm{i}_n}}(\hat{\mathbf{p}}^k)\mathbf{d} \leq 0 \text{ for all active constraints } c_{\mathrm{i}_n} \text{ and } |d_i| \leq 1,\ i = 1, \dots, n_\mathrm{p}.$$

However, the nonlinearity of the active constraints and the fact that only active constraints are taken into account raise convergence difficulties. These difficulties can be avoided by taking all constraints into account and forcing $\mathbf{d}_k$ to point towards the interior of the feasible set $\mathbb{P}$ when constraints are active at $\hat{\mathbf{p}}^k$ (Luenberger, 1973; Minoux, 1983). The feasible direction $\mathbf{d}_k$ is then computed as in Section 4.3.4.6, and

$$\lambda_k = \arg \min_{\lambda \in \mathbb{\Lambda}} j(\hat{\mathbf{p}}^k + \lambda \mathbf{d}_k),$$

where

$$\mathbb{\Lambda} = \{\lambda \geq 0 \mid \mathbf{c}_\mathrm{i}(\hat{\mathbf{p}}^k + \lambda \mathbf{d}_k) \leq \mathbf{0}\}.$$

The stopping rule at Step 3 is now based on the value obtained for α when solving the linear program involved in the computation of $\mathbf{d}_k$, *i.e.*

Step 3': If $\alpha = 0$ then stop; else ($\alpha < 0$) set $\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^{k+}$, increment k by one and go to Step 2.

Various modifications of the algorithm have been suggested to facilitate the determination of $\mathbf{d}_k$. The basic idea is not to take into account those constraints that are far enough from being saturated.

The method also applies when equality constraints $\mathbf{c}_e(\mathbf{p}) = \mathbf{0}$ are present. When $\mathbf{c}_e(\mathbf{p})$ is affine in $\mathbf{p}$, *i.e.* $\mathbf{c}_e(\mathbf{p}) = \mathbf{A}\mathbf{p} + \mathbf{b}$, it suffices to take the additional constraint $\mathbf{A}\mathbf{d} = \mathbf{0}$ into account when computing $\mathbf{d}_k$. When $\mathbf{c}_e(\mathbf{p})$ is not affine, a penalty function may be used (Polak, 1971).

So far, only the first derivatives of j and $\mathbf{c}_i$ have been considered. Incorporation of second derivatives leads to *second-order methods of feasible directions* (Polak, 1971). The computation of $\mathbf{d}_k$ is then performed by minimization of a quadratic function under quadratic constraints (Lawson and Hanson, 1974).

4.3.5 Non-differentiable cost functions

Except for one-dimensional search, the methods considered so far have dealt with cost functions differentiable with respect to $\mathbf{p}$. Non-differentiable cost functions, such as the L_1 norm used in least-modulus estimation or the L_∞ norm used in minimax estimation, require specific methods; see also Section 4.3.9.2.

As a motivating example, consider the following cost function (Polyak, 1987)

$$j(\mathbf{p}) = |p_1 - p_2| + 0.2|p_1 + p_2|, \quad \mathbf{p} = (p_1,\ p_2)^{\mathrm{T}}.$$

If the initial point $\hat{\mathbf{p}}^0$ is taken as $(1,\ 1)^{\mathrm{T}}$, the cost increases in the directions of both coordinate axes, so Powell's method does not apply, even if it does not use the gradient of the cost.

The three methods to be presented assume the cost function to be convex. The first extends the gradient algorithm to non-differentiable costs *via* the notion of subgradient. The second relies on iterative construction of a piecewise linear approximation of the cost. In contrast to the first, it provides monotone convergence. The third relies on an ellipsoidal algorithm, of the same type as that suggested for linear programming above. It has been successfully applied even to some non-convex problems

4.3.5.1 Subgradient method

A *subgradient* (Shor, 1985) at $\mathbf{p}$ of a convex function j is a vector $\mathbf{a}$ that satisfies for any $\mathbf{r}$

$$j(\mathbf{p} + \mathbf{r}) \geq j(\mathbf{p}) + \mathbf{a}^{\mathrm{T}}\mathbf{r}.$$

When j is differentiable at $\mathbf{p}$, the subgradient $\tilde{\mathbf{g}}(\mathbf{p}) = \mathbf{g}(\mathbf{p})$ is unique. Otherwise, the *subdifferential* of j at $\mathbf{p}$, *i.e.* the set of all its subgradients $\tilde{\mathbf{g}}(\mathbf{p})$, will be denoted by $\partial j(\mathbf{p})$; see Figure 4.37. Any subgradient has the following properties (Polyak, 1987):

— if j is convex, the set of all its subgradients is bounded over any set defined by $\{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid j(\mathbf{p}) \leq \alpha\}$;

— if j is convex, $[\tilde{\mathbf{g}}(\mathbf{p}_2) - \tilde{\mathbf{g}}(\mathbf{p}_1)]^{\mathrm{T}}(\mathbf{p}_2 - \mathbf{p}_1) \geq 0$ (the subgradient is a monotone operator);

— $\hat{\mathbf{p}}$ is a minimizer of the unconstrained convex cost function j if and only if $\mathbf{0} \in \partial j(\hat{\mathbf{p}})$.

— if $g(\mathbf{p}_1, \mathbf{p}_2)$ is the directional derivative of j at $\mathbf{p}_1$ towards $\mathbf{p}_2$,

$$g(\mathbf{p}_1, \mathbf{p}_2) = \lim_{\varepsilon \to 0} \frac{j(\mathbf{p}_1 + \varepsilon \mathbf{p}_2) - j(\mathbf{p}_1)}{\varepsilon},$$

(also called the Fréchet derivative), then

$$g(\mathbf{p}_1, \mathbf{p}_2) = \max_{\mathbf{a} \in \partial j(\mathbf{p}_1)} \mathbf{a}^T \mathbf{p}_2.$$

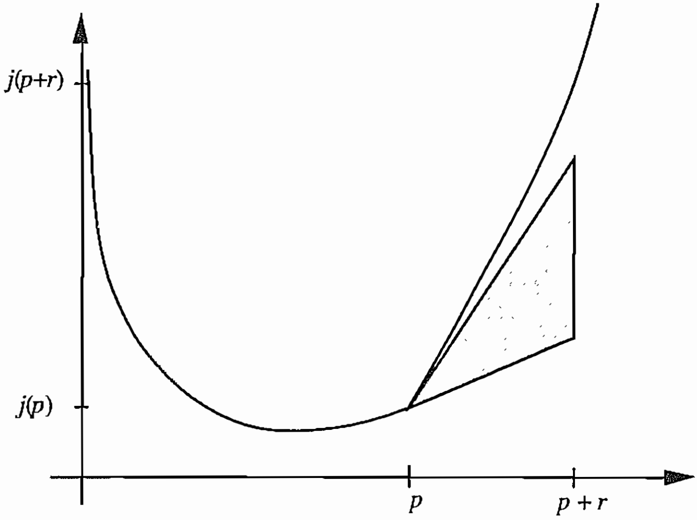


Figure 4.37. Subdifferential $\partial j(p)$ of a convex function

The following elementary rules allow a simple computation of a subgradient of a convex function j.

— Let j_i ($i = 1, \ldots, m$) be convex functions, with respective subgradients $\tilde{\mathbf{g}}_i$. If j is defined by

$$j(\mathbf{p}) = \sum_{i=1}^{m} \alpha_i j_i(\mathbf{p}),$$

then

$$\tilde{\mathbf{g}}(\mathbf{p}) = \sum_{i=1}^{m} \alpha_i \tilde{\mathbf{g}}_i(\mathbf{p})$$

is a subgradient of j at $\mathbf{p}$. If j is defined by

$$j(\mathbf{p}) = \max_{1 \le i \le m} j_i(\mathbf{p}),$$

then $\partial j(\mathbf{p})$ is the convex hull of

$$\bigcup_{i\in\mathbb{I}(\mathbf{p})} \{\partial j_i(\mathbf{p})\}, \text{ with } \mathbb{I}(\mathbf{p}) = \{i \in \{1, \dots, m\} \mid j_i(\mathbf{p}) = j(\mathbf{p})\}.$$

— Let $\mathbf{A}$ be an $m \times n_\mathrm{p}$ matrix, and f a function over $\mathbb{R}^m$ such that $j(\mathbf{p}) = f(\mathbf{Ap})$; then a subgradient of j at $\mathbf{p}$ is $\mathbf{A}^\mathrm{T}\tilde{\mathbf{g}}_\mathrm{f}(\mathbf{p})$, with $\tilde{\mathbf{g}}_\mathrm{f}$ a subgradient of f.

The subgradient method derives from the gradient algorithm, and corresponds to

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \lambda_k \tilde{\mathbf{g}}(\hat{\mathbf{p}}^k).$$

It is not always possible to make the cost decrease in the direction $-\tilde{\mathbf{g}}$ suggested by the subgradient. Consider again, for example, the cost function

$$j(\mathbf{p}) = |p_1 - p_2| + 0.2|p_1 + p_2|,$$

with a subgradient at $\hat{\mathbf{p}}^0 = (1, 1)^\mathrm{T}$ given by $\tilde{\mathbf{g}}(\hat{\mathbf{p}}^0) = (1.2, -0.8)^\mathrm{T}$. It is easy to check that j increases along $-\tilde{\mathbf{g}}(\hat{\mathbf{p}}^0)$. It is therefore meaningless to look for any optimal step size λ_k. It is equally impossible to keep λ constant. Consider for example the one-parameter cost function $j(p) = |p|$. Since $|\tilde{g}(p)| = 1$ for any nonzero p, a constant step size λ would imply $|\hat{p}^{k+1} - \hat{p}^k| = \lambda$ for any k. The sequence λ_k must be chosen *a priori*, and it can be shown that, for a convex cost function, a sequence satisfying

$$\lambda_k \to 0 \text{ as } k \to \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \lambda_k = \infty,$$

ensures convergence of $\hat{\mathbf{p}}^k$ towards the optimum. The second condition, however, imposes slow convergence, so the method is not recommended in practice (Polyak, 1987).

REMARK 4.22

An approximate subgradient, easier to compute than a subgradient, may also be employed (Polyak, 1987). ◊

4.3.5.2 Cutting-plane method

For any convex cost function j, from the definition of the subgradient, we have

$$j(\hat{\mathbf{p}}) \geq j(\hat{\mathbf{p}}^k) + \tilde{\mathbf{g}}^\mathrm{T}(\hat{\mathbf{p}}^k)(\hat{\mathbf{p}} - \hat{\mathbf{p}}^k),$$

so any minimizer $\hat{\mathbf{p}}$ of j over some prior polytope $\mathbb{P}$ is in

$$\mathbb{P}_k = \{\mathbf{p} \in \mathbb{P} \mid \tilde{\mathbf{g}}^\mathrm{T}(\hat{\mathbf{p}}^i)(\mathbf{p} - \hat{\mathbf{p}}^i) \leq j(\hat{\mathbf{p}}) - j(\hat{\mathbf{p}}^i),\ i = 1, \dots, k\}.$$

If the value of $j(\hat{\mathbf{p}})$ is unknown, this set can be replaced by

$$\mathbb{P}_k = \{\mathbf{p} \in \mathbb{P} \mid \tilde{\mathbf{g}}^{\mathrm{T}}(\hat{\mathbf{p}}^i)(\mathbf{p} - \hat{\mathbf{p}}^i) \leq 0,\ i = 1, \dots, k\}.$$

In both cases, $\mathbb{P}_k$ is also a polytope in $\mathbb{R}^{n_\mathrm{p}}$. The next point $\hat{\mathbf{p}}^{k+1}$ must be chosen so as to reduce the size of $\mathbb{P}_{k+1}$ as much as possible. In the cutting-plane method (Kelley, 1960), $\hat{\mathbf{p}}^{k+1}$ minimizes a linear approximation of j built on the basis of the previous observations. It is obtained by solving the linear-programming problem:

$$\hat{\mathbf{p}}^{k+1} = \arg\min_{\mathbf{p}} \alpha(\mathbf{p}),$$

under the constraints

$$\mathbf{p} \in \mathbb{P} \quad \text{and} \quad j(\hat{\mathbf{p}}^i) + \tilde{\mathbf{g}}^{\mathrm{T}}(\hat{\mathbf{p}}^i)(\mathbf{p} - \hat{\mathbf{p}}^i) \leq \alpha(\mathbf{p}),\ i = 1, \dots, k.$$

Consider the extended vector $\mathbf{p}_\mathrm{e} = (\mathbf{p}^{\mathrm{T}},\ \alpha)^{\mathrm{T}}$ in $\mathbb{R}^{n_\mathrm{p}+1}$. From the prior polytope $\mathbb{P}_0 = \mathbb{P}$ containing $\hat{\mathbf{p}}^0$ and $\hat{\mathbf{p}}$ and bounds on α ($\alpha_{\min} < j(\hat{\mathbf{p}})$, $\alpha_{\max} \geq j(\hat{\mathbf{p}}^0)$), an initial polytope can be built that contains $\hat{\mathbf{p}}_\mathrm{e}^0$. The updating of the polytope after each new evaluation of the cost corresponds to one iteration of the algorithm for exact polyhedral description in Section 5.4.1.3. Let $\mathbf{p}_\mathrm{e}^+ = (\mathbf{p}^{+\mathrm{T}},\ \alpha^+)^{\mathrm{T}}$ be a vertex of the updated polytope such that α^+ is the minimal value taken by α over all vertices. The point $\hat{\mathbf{p}}^{k+1}$ is chosen as $\mathbf{p}^+$.

The method is guaranteed to converge for a convex cost function. When j is not convex, it can still be applied iteratively, initializing each iteration with a reduced-size polytope (which may not contain $\hat{\mathbf{p}}$) centred on the best value of $\mathbf{p}$ obtained at the previous iteration.

REMARK 4.23

Another polyhedral approach, operating now in $\mathbb{R}^{n_\mathrm{p}}$, is to choose an interior point of the polytope $\mathbb{P}_k$ as $\hat{\mathbf{p}}^{k+1}$, for instance the barycentre of its vertices or the centre of the minimum-volume axis-aligned orthotope containing $\mathbb{P}^k$. This method is very close to the method of centres presented in Section 4.3.4.6. ◊

4.3.5.3 Ellipsoidal method

The previous approach requires solution of a linear-programming problem, for which an ellipsoidal algorithm can be employed (Sections 4.3.4.1 and 5.4.1.1). Consider the case where the value of $j(\hat{\mathbf{p}})$ is unknown. The minimizer $\hat{\mathbf{p}}$ is in

$$\mathbb{P}_{k-1} = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid \tilde{\mathbf{g}}^{\mathrm{T}}(\hat{\mathbf{p}}^i)(\mathbf{p} - \hat{\mathbf{p}}^i) \leq 0,\ i = 1, \dots, k-1\}.$$

Provided that this domain is bounded, it is possible to construct, recursively, an ellipsoid $\mathbb{E}_k$ guaranteed to contain it:

$$\mathbb{E}_k = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid (\mathbf{p} - \hat{\mathbf{p}}^k)^{\mathrm{T}}\mathbf{M}_k^{-1}(\mathbf{p} - \hat{\mathbf{p}}^k) \leq 1\}.$$

The point $\hat{\mathbf{p}}^{k+1}$ is then chosen as the centre of the smallest-volume ellipsoid that contains $\mathbb{E}_k \cap \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid \tilde{\mathbf{g}}^{\mathrm{T}}(\hat{\mathbf{p}}^k)(\mathbf{p} - \hat{\mathbf{p}}^k) \leq 0\}$ (Bland, Goldfarb and Todd, 1981). The algorithm is:

Step 0: Choose $\hat{\mathbf{p}}^0$, $\mathbf{M}_0 = \rho\mathbf{I}_{n_\mathrm{p}}$, $\rho >> 1$, such that $\mathbb{E}_0$ contains the minimizer $\hat{\mathbf{p}}$. Set $k = 0$.

Step 1: If $\tilde{\mathbf{g}}^\mathrm{T}(\hat{\mathbf{p}}^k)\mathbf{M}_k\tilde{\mathbf{g}}(\hat{\mathbf{p}}^k) = 0$ (or < 0 because of numerical problems), stop. Else set

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k - \tau\frac{\mathbf{M}_k\tilde{\mathbf{g}}(\hat{\mathbf{p}}^k)}{\sqrt{\tilde{\mathbf{g}}^\mathrm{T}(\hat{\mathbf{p}}^k)\mathbf{M}_k\tilde{\mathbf{g}}(\hat{\mathbf{p}}^k)}},$$

$$\mathbf{M}_{k+1} = \delta\left[\mathbf{M}_k - \sigma\frac{\mathbf{M}_k\tilde{\mathbf{g}}(\hat{\mathbf{p}}^k)\tilde{\mathbf{g}}^\mathrm{T}(\hat{\mathbf{p}}^k)\mathbf{M}_k}{\tilde{\mathbf{g}}^\mathrm{T}(\hat{\mathbf{p}}^k)\mathbf{M}_k\tilde{\mathbf{g}}(\hat{\mathbf{p}}^k)}\right],$$

with

$$\tau = \frac{1}{n_\mathrm{p} + 1}, \sigma = \frac{2}{n_\mathrm{p} + 1} \quad \text{and} \quad \delta = \frac{n_\mathrm{p}^2}{n_\mathrm{p}^2 - 1}.$$

Step 2: Increment k by one and go to Step 1.

A stopping rule based on the decrease of $j(\hat{\mathbf{p}}^k)$ between two iterations could be introduced, in addition to the numerical stability test of Step 1. This method may be considered a space-dilation method (one might also say a variable-metric method, see Section 4.3.3.6), because it corresponds to a subgradient method with a change of metric in the direction of $\tilde{\mathbf{g}}(\hat{\mathbf{p}}^k)$.

REMARKS 4.24

— If the problem involves linear inequality constraints on $\mathbf{p}$,

$$\mathbf{p} \in \mathbb{Q}_m = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid \mathbf{a}_i^\mathrm{T}\mathbf{p} \leq b_i, \ i = 1, \ldots, m\},$$

a point $\hat{\mathbf{p}}^{k+1} \in \mathbb{Q}_m$ can be chosen by the same type of ellipsoidal algorithm (Section 4.3.4.1). If at iteration k some of these constraints are violated by $\hat{\mathbf{p}}^k$, the subgradient $\tilde{\mathbf{g}}(\hat{\mathbf{p}}^k)$ is replaced by the vector $\mathbf{a}_j$ associated with one of them, for instance the most violated.

— If the inequality constraints are nonlinear, the same approach can still be used, with the $\mathbf{a}_i$'s replaced by the gradients of the constraints at $\hat{\mathbf{p}}^k$. For a comparison between this ellipsoidal approach and more classical methods of nonlinear programming on both convex and nonconvex problems, see (Ecker and Kupferschmid, 1985). The ellipsoidal method turns out to be relatively insensitive to a lack of precision in the evaluation of the cost and to the choice of the initial value for the parameters, and very efficient during the initial phase of the optimization. ◊

4.3.5.4 Application to L_1 estimation

This section illustrates a possible application of the previous algorithms. For other specific methods, see (Dodge, 1987; Gonin and Money, 1989). Note that replacing an L_1 estimator by a Huber non-redescending M-estimator (Section 3.7.3), with a small threshold δ, would make the optimization problem differentiable, so the local methods of Section 4.3.3 would apply.

The cost function to be minimized can be written as

$$j(\mathbf{p}) = \sum_{i=1}^{n_t} w_i\, |y(t_i) - y_m(t_i, \mathbf{p})|.$$

The algorithm proposed by *Osborne and Watson* (1971) is based on the sequential solution of convex problems obtained by linearizing $y_m(t, \mathbf{p})$.

Step 0: Choose $\hat{\mathbf{p}}^0$, set $k = 0$.
Step 1: Compute

$$\delta\hat{\mathbf{p}}^k = \arg\min_{\delta\mathbf{p}} \sum_{i=1}^{n_t} w_i\, \left|y(t_i) - y_m(t_i, \hat{\mathbf{p}}^k) - \frac{\partial y_m(t_i, \mathbf{p})}{\partial \mathbf{p}^T}\right|_{\hat{\mathbf{p}}^k} \delta\mathbf{p}|.$$

Step 2: Compute

$$\lambda_k = \arg\min_{\lambda>0} \sum_{i=1}^{n_t} w_i\, |y(t_i) - y_m(t_i, \hat{\mathbf{p}}^k + \lambda\delta\hat{\mathbf{p}}^k)|.$$

Step 3: Set $\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k + \lambda_k\delta\hat{\mathbf{p}}^k$, increment k by one and go to Step 1.

The computation of $\delta\hat{\mathbf{p}}^k$ (Step 1) can be performed by the cutting-plane method of Section 4.3.5.2 or the ellipsoidal method of Section 4.3.5.3. The computation of λ_k (Step 2) corresponds to a one-dimensional minimization (Section 4.3.2). A stopping rule (based, *e.g.*, on $|j(\hat{\mathbf{p}}^{k+1}) - j(\hat{\mathbf{p}}^k)|$) must be introduced at Step 3.

EXAMPLE 4.18

Consider the model response described by

$$y_m(\boldsymbol{\xi}, \mathbf{p}) = p_1\xi_1 + p_2\xi_2 + p_1^3(1 - \xi_1) + p_2^2(1 - \xi_2),$$

where the vector $\boldsymbol{\xi}$ characterizes the experimental conditions. Assume that the experimental conditions are

$$\boldsymbol{\xi}^1 = (1, 0)^T, \quad \boldsymbol{\xi}^2 = \boldsymbol{\xi}^3 = (1, 1)^T.$$

The parameters are to be estimated in the sense of the (unweighted) least-modulus criterion, so the cost function to be minimized is

$$j(\mathbf{p}) = |p_1 + p_2^2 - y_1| + |p_1 + p_2 - y_2| + |p_1 + p_2 - y_3|.$$

The Osborne and Watson algorithm was used, using the ellipsoidal approach of Section 4.3.5.3 at Step 1 and Brent's derivative-free one-dimensional optimization method at Step 2. Figure 4.38 presents the cost contours when the measurements on the process are $\mathbf{y}^s = (5, 2, 4)^T$.

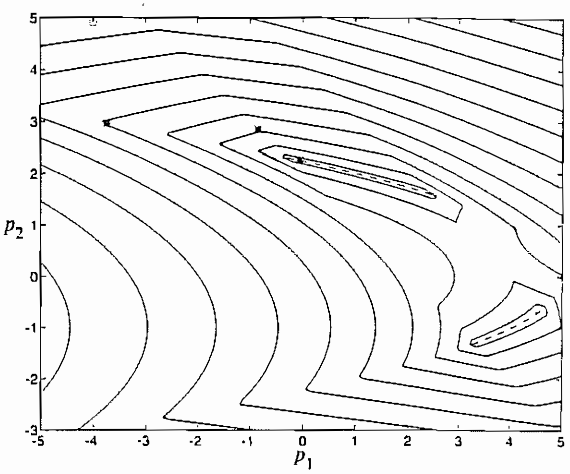


Figure 4.38. Cost contours for the L_1 optimization problem of Example 4.18

All values of $\mathbf{p}$ such that $p_1 + p_2^2 = 5$ and $2 \leq p_1 + p_2 \leq 4$ (dashed line) are minimizers, so there is no local uniqueness of the estimate (compare with Example 4.21, where the same model structure will be considered in the context of least squares). Starting from $\hat{\mathbf{p}}^0 = (-4, 4.9)^{\mathrm{T}}$, which corresponds to $j(\hat{\mathbf{p}}^0) = 19.21$, after 50 evaluations of the cost, 4 optimizations of the approximate cost corresponding to Step 1 (at $\hat{\mathbf{p}}^0$, indicated by a circle, then at the points indicated by stars), and 42 evaluations of subgradients, the algorithm finds $\hat{\mathbf{p}} = (-0.0706, 2.2518)^{\mathrm{T}}$, which corresponds to $j(\hat{\mathbf{p}}) = 2$. All classical methods for local optimization that are not designed to handle nondifferentiability just rush to a point where the cost is not differentiable and fail piteously. ◊

4.3.6 Initialization

Most methods considered so far take advantage of the local properties of the cost function j. Thus at best they converge to a *local* optimum. Nothing guarantees, in general, that there are no other feasible values of $\mathbf{p}$ yielding lower $j(\mathbf{p})$. It is therefore important to choose an initial value $\hat{\mathbf{p}}^0$ as close as possible to a global optimizer $\check{\mathbf{p}}$ of j. We have seen, for instance, that it is often possible to make the model output linear in the parameters so as to find $\hat{\mathbf{p}}^0$ by the least-squares method (Examples 4.3, 4.4 and 4.6).

It is of course also advisable to start other local optimizations from points picked at random in the prior feasible space $\mathbb{P}$, to see whether the algorithm always converges towards a similar value of $\mathbf{p}$. Otherwise, one should rank the various candidate optimizers according to the value of the cost, and keep the best. One may then realize that quite different values of $\mathbf{p}$ yield an optimal or quasi-optimal value of the cost. The uncertainty in the estimated parameters will then obviously be very large.

A possible method is as follows:

— choose a minimization technique that guarantees convergence to the local minimum located in the same basin of attraction as $\hat{\mathbf{p}}^0$,

— perform a large number of local minimizations, picking $\hat{\mathbf{p}}^0$ at random in $\mathbb{P}$ according to a uniform distribution.

Provided that the basins of attraction of the global minimizers are not too small, one will thus locate all global minimizers. This is a first example of a global optimization method, with the ability to escape local minima. Others, which also aim to solve, at least partially, the problems raised by initialization, will be presented in Section 4.3.9.

4.3.7 Termination

Whenever an iterative minimization technique is used, one has to decide when to stop the process. The most commonly used stopping conditions are

$$j(\hat{\mathbf{p}}^k) < \delta,$$

$$|j(\hat{\mathbf{p}}^{k+1}) - j(\hat{\mathbf{p}}^k)| < \delta,$$

$$|[j(\hat{\mathbf{p}}^{k+1}) - j(\hat{\mathbf{p}}^k)]/j(\hat{\mathbf{p}}^k)| < \delta,$$

$$k = k_{\max}.$$

By itself, none of these conditions is usually satisfactory. One is often at a loss how to choose the threshold δ in the first three. The last condition requires advance knowledge of how many iterations will be required, which is seldom realistic. Even if this type of condition is always introduced as a safety measure to avoid infinite loops, $k_{\max}$ is then only an indication of the maximum effort one is prepared to allocate to this local minimization.

If the cost function is differentiable, any minimizer $\hat{\mathbf{p}}$ which is neither at infinity nor on a constraint corresponds to a stationary point and satisfies

$$\frac{\partial j}{\partial \mathbf{p}}\Big|_{\mathbf{p}=\hat{\mathbf{p}}} = \mathbf{0}.$$

From a *mathematical stand-point*, one would like to stop iterating when this equation is satisfied. In practice, however, only an approximate computer representation of the gradient is available. A more realistic course is to stop when the gradient no longer differs significantly from zero, *i.e.* when its mantissa no longer contains any significant digit.

Three types of error are introduced by numerical computation, affecting the accuracy of the results:

— *truncation errors* (or *rounding errors*, depending on the computer), resulting from finite word length; truncation error arises because a computer number x stands for all numbers between x and x plus one least significant bit (LSB);

— *computation errors*, mainly introduced when two very close numbers are subtracted;

— *methodological errors*, not considered here.

To estimate the number of significant digits in a result obtained by computer, the CESTAC method (*Contrôle et Estimation STochastique des Arrondis de Calcul*) (Vignes, 1978, 1988; Pichat and Vignes, 1993) may be used. The idea is to study the statistics of all computer results that may equally well represent the mathematical result sought, here the gradient of the cost function at $\hat{\mathbf{p}}^k$.

To take truncation or rounding errors into account, the computations are performed several times, randomly adding or subtracting LSB's in the mantissas of the results of all numerical operations on reals. If prob(i) denotes the probability of adding i LSB's to any result, then

$$\text{prob}(0) = \text{prob}(1) = 0.5$$

when dealing with truncation errors, and

$$\text{prob}(0) = 0.5,\ \ \text{prob}(-1) = \text{prob}(1) = 0.25$$

for rounding errors.

To take computation errors into account, early versions of CESTAC also randomly permuted the order of additions in sums of products (Vignes, 1978), which led to a much more complicated implementation.

Consider one component of the gradient thus evaluated. If the population of the results obtained is distributed as in Figure 4.39, the gradient will be considered as differing significantly from zero, and further iterations will be authorized. If, on the other hand, all components of the gradient are distributed as in Figure 4.40, one cannot conclude that the gradient differs from zero, so no information is available on the direction to be followed, and iteration should be terminated.

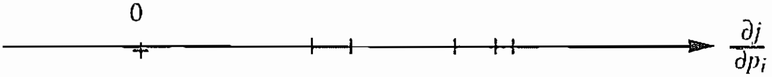


Figure 4.39. The ith component of the gradient can be considered to differ from zero

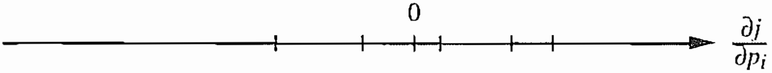


Figure 4.40. It is unclear whether the ith component of the gradient differs from zero.

To quantify these ideas, assume that the set of the results r_i, $i = 1, \ldots, n$, obtained when computing an actual quantity r consists of samples from a Gaussian population. Then

$$\text{prob}\left(r \in \left[\mu - t_{\alpha/2}\frac{\sigma}{\sqrt{n}}, \mu + t_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right]\right) = 1 - \alpha,$$

where

$$\mu = \frac{1}{n}\sum_{i=1}^{n} r_i\ ,\ \ \sigma^2 = \frac{1}{n-1}\ \sum_{i=1}^{n}(r_i - \mu)^2,$$

and $t_{\alpha/2}$ is the (tabulated) value with probability α of being exceeded by the absolute value of a random variable having a Student t-distribution with $n-1$ degrees of freedom. The number of significant digits of any of the results r_i is then estimated by

$$\hat{n}_{\mathrm{sd}_\mathrm{r}} = \log_{10}\frac{|\mu|}{(\frac{2t_{\alpha/2}\sigma}{\sqrt{n}})} = \log_{10}\frac{|\mu|}{\sigma} - \log_{10}(2\frac{t_{\alpha/2}}{\sqrt{n}}),$$

and the number of significant digits of their mean μ by

$$\hat{n}_{\mathrm{sd}_\mathrm{m}} = \log_{10}\frac{|\mu|}{\sigma} - \log_{10}(\frac{t_{\alpha/2}}{\sqrt{n}}).$$

Perturbed computations are usually performed only three times ($n = 3$). Then, for $\alpha = 0.05$,

$$\hat{n}_{\mathrm{sd}_\mathrm{m}} = \log_{10}\frac{|\mu|}{\sigma} - 0.4.$$

If $\hat{n}_{\mathrm{sd}_\mathrm{m}}$ is less than one for all components of the gradient, iterative local minimization is terminated.

REMARKS 4.25

— The LSB depends on the computer and the precision used. The number of the iterations actually performed will also depend on these factors. The more accurate the computation, the longer iterative procedures can proceed significantly. In contrast to the stopping conditions at the beginning of this section, this condition has a rational basis.

— Contrary to what is assumed here, the population of possible results may be far from Gaussian, especially if conditional branching is involved and the decision taken differs between realizations. A way to mitigate this difficulty is to use a synchronous implementation of CESTAC to detect any intermediate result, such as the result of a numerical test, that becomes insignificant.

— Although the gradient must be evaluated several times at each iteration, the computational load may turn out to be lighter than with a traditional test, because CESTAC often terminates iterations much earlier.

— Various software tools are available to implement CESTAC in FORTRAN or ADA code automatically (Pichat and Vignes, 1993). ◊

4.3.8 Recursive techniques

As with least squares, one may wish to treat data one by one, indexed by an integer variable i, instead of considering all of them as a batch. Assume that there exists a prediction error $e_\mathrm{p}(t_i, \mathbf{p})$ (possibly an output error) that becomes a sequence of independent random variables $\varepsilon(t_i)$ with probability density $\pi_\varepsilon[e_\mathrm{p}(t_i, \mathbf{p}^*)]$ when the parameters take their true value. Note that $t_{i+1} - t_i$ may depend on i. Moreover, t_i does not necessarily refer to time, and may simply correspond to the ith observation of the process. The maximum-likelihood approach minimizes the cost

$$j_{\mathrm{ml}}(\mathbf{p}) = -\frac{1}{n_t}\sum_{i=1}^{n_t} \ln \pi_\varepsilon[e_\mathrm{p}(t_i, \mathbf{p})].$$

Under the hypothesis of ergodicity, $j_{\mathrm{ml}}(\mathbf{p})$ can be seen as an approximation of the cost

$$j(\mathbf{p}) = -\underset{e_\mathrm{p}(\mathbf{p})}{\mathrm{E}}\{\ln \pi_\varepsilon[e_\mathrm{p}(\mathbf{p})]\}.$$

With the help of *stochastic-approximation techniques*, (Robbins and Monro, 1951; Dvoretzky, 1956; Polyak and Tsypkin, 1973; Saridis, 1974; Ermoliev and Wets, 1988), it is possible to minimize j without ever evaluating any mathematical expectation. As a fringe benefit, one gets a recursive algorithm, even when the model output depends nonlinearly on the parameters. One may, for example, use a *stochastic gradient* algorithm, derived from the gradient algorithm for the cost function j, *i.e.*

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k + \lambda_k \frac{\partial}{\partial \mathbf{p}} \underset{e_\mathrm{p}(\mathbf{p})}{\mathrm{E}}\{\ln \pi_\varepsilon[e_\mathrm{p}(\mathbf{p})]\}\Big|_{\mathbf{p}=\hat{\mathbf{p}}^k},$$

by replacing evaluation of the mathematical expectation by random picking of a prediction error from the set of all those that could have been obtained. For this purpose, the prediction error associated with the measurement $y(t_{k+1})$ is used, to get

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k + \lambda_k \frac{\partial}{\partial \mathbf{p}} \ln \pi_\varepsilon[e_\mathrm{p}(t_{k+1}, \mathbf{p})]\Big|_{\mathbf{p}=\hat{\mathbf{p}}^k},$$

that is

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k + \lambda_k \frac{\frac{\partial}{\partial \mathbf{p}}\pi_\varepsilon[e_\mathrm{p}(t_{k+1}, \mathbf{p})]}{\pi_\varepsilon[e_\mathrm{p}(t_{k+1}, \mathbf{p})]}\Big|_{\mathbf{p}=\hat{\mathbf{p}}^k}.$$

The gain λ_k must satisfy three conditions:

— $\lambda_k > 0$ (the steps are in the right direction),

— $\sum\limits_{k=0}^{\infty} \lambda_k = \infty$ (all feasible parameter vectors can be reached),

— $\sum\limits_{k=0}^{\infty} \lambda_k^2 < \infty$ (the influence of the noise disappears asymptotically).

A possible gain is therefore $\lambda_k = \lambda_0/(k+1)$. See also the notion of averaging below, and Section 6.4.3.2 for strategies with performance less sensitive to the choice of λ_0.

As with the recursive least-squares algorithm without forgetting, the correction gain tends to zero as the number of iterations tends to infinity. The associated estimator is consistent, like the maximum-likelihood estimator, but unlike it is not asymptotically efficient. For a general study of the convergence properties of this type of algorithm, see (Benveniste, Métivier and Priouret, 1987, 1990).

A *stochastic Newton* algorithm can also be used (Ljung and Söderström, 1983; Tsypkin, 1983; Tsypkin and Lototsky, 1985), defined as

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^{k} + \mathbf{F}_{k+1}^{-1}(\hat{\mathbf{p}}^{k}) \frac{\partial}{\partial \mathbf{p}} \ln \pi_{\varepsilon}[e_{\mathrm{p}}(t_{k+1}, \mathbf{p})]_{|\mathbf{p}=\hat{\mathbf{p}}^{k}},$$

where $\mathbf{F}_k$ is the Fisher information matrix for the first k data points, which can be written, as will be seen in Chapter 5,

$$\mathbf{F}_k(\mathbf{p}) = - \underset{\mathbf{e}_{\mathrm{p}}|\mathbf{p}}{\mathrm{E}} \left\{ \frac{\partial^2}{\partial \mathbf{p} \partial \mathbf{p}^{\mathrm{T}}} \ln \pi_{\varepsilon}[\mathbf{e}_{\mathrm{p}}(\mathbf{p})] \right\},$$

with

$$\mathbf{e}_{\mathrm{p}}(\mathbf{p}) = [e_{\mathrm{p}}(t_1, \mathbf{p}), \ldots , e_{\mathrm{p}}(t_k, \mathbf{p})]^{\mathrm{T}}.$$

Provided that the hypotheses made in Section 3.3.3 when presenting the properties of maximum-likelihood estimators are satisfied, this algorithm is asymptotically efficient. Note that it still involves the evaluation of a mathematical expectation, in contrast to the stochastic gradient algorithm. We shall, however, see in Chapter 5 how to obtain an analytical expression for $\mathbf{F}_k(\mathbf{p})$, which makes the complexity of *one iteration* of the stochastic Newton algorithm equivalent to that of *one iteration* of a conventional Newton algorithm. If we define the average Fisher information matrix per sample as

$$\mathbf{F}_{k_{\mathrm{ps}}}(\hat{\mathbf{p}}^{k}) = \frac{1}{k} \mathbf{F}_k(\hat{\mathbf{p}}^{k}),$$

the stochastic Newton algorithm can be written as

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^{k} + \frac{1}{k+1} \mathbf{F}_{k+1_{\mathrm{ps}}}^{-1}(\hat{\mathbf{p}}^{k}) \frac{\partial}{\partial \mathbf{p}} \ln \pi_{\varepsilon}[e_{\mathrm{p}}(t_{k+1}, \mathbf{p})]_{|\mathbf{p}=\hat{\mathbf{p}}^{k}},$$

which has a decreasing gain $\lambda_k = 1/(k+1)$.

When the random variables $\varepsilon(t)$ are i.i.d. $\mathcal{N}(0, \sigma^2)$, this algorithm is a stochastic version of the Gauss-Newton algorithm of Sections 4.2.4 and 4.3.3.4.

Since the stochastic Newton algorithm is asymptotically efficient, it provides a recursive estimator with the same *asymptotic* optimality properties as the maximum-likelihood estimator, which does not mean that the properties on a small set of data points will be similar.

Assume now that time-varying parameters are to be tracked. It is no longer possible to let the adaptation gain tend to zero without any precautions, as already mentioned for recursive least-squares. Most often, λ_k is made to tend to some constant value, allowing the algorithm to continue to take prediction errors into account. One may prefer to let λ_k tend to zero and $\hat{\mathbf{p}}^k$ converge, but the prediction error should then be monitored to check whether the adaptation gain needs to be increased. The choice will be guided by the type of parameter variation that is assumed to be possible. If the parameters vary slowly and continuously, a technique that makes λ_k tend to a nonzero constant seems preferable. If, on the other hand, the parameters jump between constant values, one should rather let

λ_k tend to zero and monitor the prediction error to detect when the gain should be inflated, see Sections 4.1.4.2 and 4.1.4.3..

EXAMPLE 4.19

Assume that for the true value of the parameters, the prediction errors become i.i.d. $\mathcal{N}(0, \sigma^2)$. As will be seen in Chapter 5, the Fisher information matrix associated with the first k data points is then given by

$$\mathbf{F}_k(\hat{\mathbf{p}}^k) = \frac{1}{\sigma^2} \sum_{i=1}^{k} \frac{\partial}{\partial \mathbf{p}} e_{\mathrm{p}}(t_i, \mathbf{p}) \frac{\partial}{\partial \mathbf{p}^{\mathrm{T}}} e_{\mathrm{p}}(t_i, \mathbf{p})_{|\mathbf{p}=\hat{\mathbf{p}}^k}.$$

In the special case where the prediction error is affine in the parameters, $\mathbf{F}_k$ does not depend on $\hat{\mathbf{p}}^k$ but only on k, so it can be computed recursively. One can indeed write

$$e_{\mathrm{p}}(t_{k+1}, \mathbf{p}) = y(t_{k+1}) - y_{\mathrm{m}}(t_{k+1}, \mathbf{p}) = y(t_{k+1}) - \mathbf{r}^{\mathrm{T}}(k)\mathbf{p}$$

and

$$\mathbf{F}_{k+1} = \mathbf{F}_k + \frac{1}{\sigma^2} \frac{\partial}{\partial \mathbf{p}} e_{\mathrm{p}}(t_{k+1}, \mathbf{p}) \frac{\partial}{\partial \mathbf{p}^{\mathrm{T}}} e_{\mathrm{p}}(t_{k+1}, \mathbf{p}),$$

so

$$\mathbf{F}_{k+1} = \mathbf{F}_k + \frac{1}{\sigma^2} \mathbf{r}(k)\mathbf{r}^{\mathrm{T}}(k).$$

The matrix-inversion lemma can then be used to compute $\mathbf{F}_{k+1}^{-1}$ without inverting a matrix at each iteration. The resulting algorithm corresponds to recursive least squares. A stochastic-gradient algorithm would in this case lead to

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k + \lambda_k \mathbf{r}(k) e_{\mathrm{p}}(t_{k+1}, \hat{\mathbf{p}}^k),$$

which is called the *Least Mean Squares* algorithm, widely used in signal processing (Widrow and Stearns, 1985; Macchi, 1995). See also Remark 4.4.

Assume now that the prediction error is no longer affine in $\mathbf{p}$. Defining an approximation $\mathbf{F}_k^{\mathrm{a}}$ of $\mathbf{F}_k(\hat{\mathbf{p}}^k)$ recursively by

$$\mathbf{F}_{k+1}^{\mathrm{a}} = \mathbf{F}_k^{\mathrm{a}} + \frac{1}{\sigma^2}\left[\frac{\partial}{\partial \mathbf{p}} e_{\mathrm{p}}(t_{k+1}, \mathbf{p}) \frac{\partial}{\partial \mathbf{p}^{\mathrm{T}}} e_{\mathrm{p}}(t_{k+1}, \mathbf{p})\right]_{|\mathbf{p}=\hat{\mathbf{p}}^k},$$

one can use the results of the affine case. Taking advantage once again of the matrix-inversion lemma gives

$$(\mathbf{F}_{k+1}^{\mathrm{a}})^{-1} = (\mathbf{F}_k^{\mathrm{a}})^{-1} - \mathbf{k}_{k+1}^{\mathrm{a}} \frac{\partial}{\partial \mathbf{p}^{\mathrm{T}}} e_{\mathrm{p}}(t_{k+1}, \mathbf{p})_{|\mathbf{p}=\hat{\mathbf{p}}^k} (\mathbf{F}_k^{\mathrm{a}})^{-1},$$

with

$$\mathbf{k}_{k+1}^{\mathrm{a}} = \frac{(\mathbf{F}_k^{\mathrm{a}})^{-1} \dfrac{\partial}{\partial \mathbf{p}} e_{\mathrm{p}}(t_{k+1}, \mathbf{p})_{|\mathbf{p}=\hat{\mathbf{p}}^k}}{\sigma^2 + \dfrac{\partial}{\partial \mathbf{p}^{\mathrm{T}}} e_{\mathrm{p}}(t_{k+1}, \mathbf{p})_{|\mathbf{p}=\hat{\mathbf{p}}^k} (\mathbf{F}_k^{\mathrm{a}})^{-1} \dfrac{\partial}{\partial \mathbf{p}} e_{\mathrm{p}}(t_{k+1}, \mathbf{p})_{|\mathbf{p}=\hat{\mathbf{p}}^k}}$$

The algorithm

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^{k} + (\mathbf{F}^{\mathrm{a}}_{k+1})^{-1} \frac{\partial}{\partial \mathbf{p}} \ln \pi_{\varepsilon}[e_{\mathrm{p}}(t_{k+1}, \mathbf{p})]_{|\mathbf{p}=\hat{\mathbf{p}}^{k}},$$

is then equivalent to

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^{k} - \frac{e_{\mathrm{p}}(t_{k+1}, \hat{\mathbf{p}}^{k})}{\sigma^2} (\mathbf{F}^{\mathrm{a}}_{k+1})^{-1} \frac{\partial}{\partial \mathbf{p}} e_{\mathrm{p}}(t_{k+1}, \mathbf{p})_{|\mathbf{p}=\hat{\mathbf{p}}^{k}},$$

and thus to

$$\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^{k} - \mathbf{k}^{\mathrm{a}}_{k+1}\, e_{\mathrm{p}}(t_{k+1}, \hat{\mathbf{p}}^{k}).$$

This corresponds to the *approximate-maximum-likelihood* method (Goodwin and Payne, 1977), similar to the recursive least-squares algorithm of Section 4.1.4 (since $\partial e_{\mathrm{p}}(t_{k+1}, \mathbf{p})/\partial \mathbf{p} = -\mathbf{r}(k)$ for LP model structures), and to the Gauss-Newton algorithm introduced in Section 4.2.4. Evidence of the superiority of this method over the extended-matrix approach presented in Section 4.2.3.2 can be found in (Ljung, Söderström and Gustavsson, 1975) . ◊

Averaging is a very promising and fascinating technique to bypass the choice of a suitable step size in the stochastic-gradient algorithm or the computation of the Fisher information matrix in the stochastic-Newton algorithm. Under some technical conditions, a stochastic-gradient-like algorithm, with the requirements on λ_k replaced by

$$\lambda_k > 0, \;\; \lambda_k \rightarrow 0, \;\; \text{and} \;\; \frac{\lambda_k - \lambda_{k+1}}{\lambda_k} = \mathrm{o}(\lambda_k),$$

which amounts to *slowing down* the algorithm, makes it possible to achieve an *optimal* asymptotic rate of convergence for the *average* of the past estimates

$$\overline{\mathbf{p}}^{k} = \frac{1}{k+1} \sum_{i=0}^{k} \hat{\mathbf{p}}^{i},$$

whatever the actual sequence $\{\lambda_k\}$ chosen (Polyak and Juditzky, 1992). The choice of this sequence becomes therefore much less critical than with the original stochastic-gradient algorithm. One may for instance use

$$\lambda_k = \frac{1}{\sqrt{k}}.$$

In practice, averaging should not start from the very beginning, where the approximation might be very bad, and a constant λ should be used until the neighbourhood of the solution is reached. Since λ_k tends to zero more slowly, the estimates $\hat{\mathbf{p}}^{k}$ move more erratically than with the usual stochastic-gradient algorithm. This is compensated for by the averaging. Kushner and Yang (1993) have proven that the approach is much more general than originally thought. The case where λ is small and constant (*e.g.* to track time-varying parameters) is considered in (Kushner and Yang, 1995).

4.3.9 Global optimization

Global optimization techniques aim to find the best possible value $\check{j}$ for the cost and the associated optimizer(s) $\check{\mathbf{p}}$, such that for any feasible $\mathbf{p}$

$$j(\mathbf{p}) \geq j(\check{\mathbf{p}}) = \check{j}.$$

Insofar as they succeed, they bypass the initialization problems raised by local methods. Many global optimization methods are available; see, *e.g.*, the books by Dixon and Szegö (1975, 1978), Hansen (1992), Horst and Tuy (1990), Mockus (1989), Ratschek and Rokne (1988), and Zhigljavsky (1991). Some are already very complex for two-parameter problems, and seem hardly generalizable to higher-dimensional problems. We shall limit ourselves to two algorithms: the first, based on random search, is extremely simple to implement but may fail to locate any global optimizer; the second, deterministic in nature, guarantees its results, at the cost of a much more complex implementation.

Note that suitable experimental conditions may eliminate parasitic local minima from parameter estimation problems, and thus make it possible to find $\check{\mathbf{p}} = \hat{\mathbf{p}}$ by local methods, as shown in the next section.

4.3.9.1 Eliminating parasitic local optima

The cost function j is inverse unimodal over $\mathbb{P}$ if and only if for every $\hat{\mathbf{p}}^0 \in \mathbb{P}$, $\hat{\mathbf{p}}^0 \neq \check{\mathbf{p}}$, there exists a path in $\mathbb{P}$ from $\hat{\mathbf{p}}^0$ to $\check{\mathbf{p}}$ along which $j(\mathbf{p})$ decreases. The global optimizer of the cost is then $\check{\mathbf{p}}$. In this section, the minimization algorithm is assumed to be such that $\hat{\mathbf{p}}^k$ will never leave $\mathbb{P}$ and $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ is assumed to be continuous in $\mathbf{p}$.

In the observation space to which $\mathbf{y}^{\mathrm{s}}$ belongs, the locus of all feasible model responses is a hypersurface $\mathbb{S}_{\mathrm{exp}} = \{\mathbf{y}^{\mathrm{m}}(\mathbf{p}) \mid \mathbf{p} \in \mathbb{P}\}$, called the *expectation surface* (Sections 5.1.1.1 and 6.4.1). Consider the case where the experiment consists of repeating only $n_{\mathrm{p}} = \dim \mathbf{p}$ distinct experiments, that is r_i observations are performed with the same experimental conditions $\boldsymbol{\xi}^i$, with

$$\sum_{i=1}^{n_{\mathrm{p}}} r_i = n_{\mathrm{t}}$$

(see also Section 6.2.2.1). Then $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ has only n_{p} distinct components, and $\mathbb{S}_{\mathrm{exp}}$ is therefore included in the hyperplane $\mathbb{H}_{\mathrm{exp}}$ containing the origin and defined by $y_i = y_j$ for all (i, j) such that $\boldsymbol{\xi}^i = \boldsymbol{\xi}^j$. Since any norm $\|.\|$ in $\mathbb{R}^{n_{\mathrm{t}}}$ is convex, *i.e.*

$$\forall\ (\mathbf{y}_1, \mathbf{y}_2) \in \mathbb{R}^{n_{\mathrm{t}}} \times \mathbb{R}^{n_{\mathrm{t}}},\ \forall\ \lambda \in [0, 1],\ \|\lambda \mathbf{y}_1 + (1 - \lambda)\mathbf{y}_2\| \leq \lambda\ \|\mathbf{y}_1\| + (1 - \lambda)\ \|\mathbf{y}_2\|,$$

the level sets of $\|\mathbf{y} - \mathbf{y}^{\mathrm{s}}\|$ are convex (balls centred at $\mathbf{y}^{\mathrm{s}}$). The intersections of these sets with $\mathbb{H}_{\mathrm{exp}}$ are thus convex, and all of them contain

$$\check{\mathbf{y}} = \arg \min_{\mathbf{y} \in \mathbb{H}_{\mathrm{exp}}} \|\mathbf{y} - \mathbf{y}^{\mathrm{s}}\|.$$

For any $\mathbf{y}^0$ in $\mathbb{H}_{\mathrm{exp}}$, $\|\mathbf{y} - \mathbf{y}^{\mathrm{s}}\|$ then decreases monotonically along the line segment from $\mathbf{y}^0$ to $\check{\mathbf{y}}$.

When $\mathbb{S}_{\mathrm{exp}}$ covers $\mathbb{H}_{\mathrm{exp}}$, and when the model structure is globally identifiable under the experimental conditions considered, there exists a unique $\check{\mathbf{p}}$ such that $\check{\mathbf{y}} = \mathbf{y}^{\mathrm{m}}(\check{\mathbf{p}})$, and to any $\mathbf{y}$ along the segment from $\mathbf{y}^{\mathrm{m}}(\hat{\mathbf{p}}^0)$ to $\check{\mathbf{y}}$ corresponds a $\mathbf{p}$ in $\mathbb{R}^{n_{\mathrm{p}}}$ such that $\mathbf{y}^{\mathrm{m}}(\mathbf{p}) = \mathbf{y}$, so $j(\mathbf{p})$ is inverse unimodal (although not necessarily convex). When $\mathbb{S}_{\mathrm{exp}}$ does not cover $\mathbb{H}_{\mathrm{exp}}$, a *sufficient* condition for the inverse unimodality of j over $\mathbb{P}$ is that $\mathbb{S}_{\mathrm{exp}}$ be convex. Note that whereas checking the convexity of $\mathbb{S}_{\mathrm{exp}}$ may turn out to be difficult for a constrained $\mathbb{P}$, the task is generally much easier when $\mathbb{P} = \mathbb{R}^{n_{\mathrm{p}}}$. When $\mathbb{S}_{\mathrm{exp}}$ is not convex, the answer depends on its shape, as illustrated by Figures 4.41 and 4.42 in the case $n_{\mathrm{p}} = 2$ for the L_∞ norm.

In Figure 4.41, if $\mathbb{S}_{\mathrm{exp}}$ is the set limited by the solid line, $j(\mathbf{p})$ is inverse unimodal over $\mathbb{P}$. Indeed, any $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ in $\mathbb{S}_{\mathrm{exp}}$ can be connected to $\check{\mathbf{y}}$ by a path contained in $\mathbb{S}_{\mathrm{exp}}$ along which $j(\mathbf{p})$ decreases monotonically.

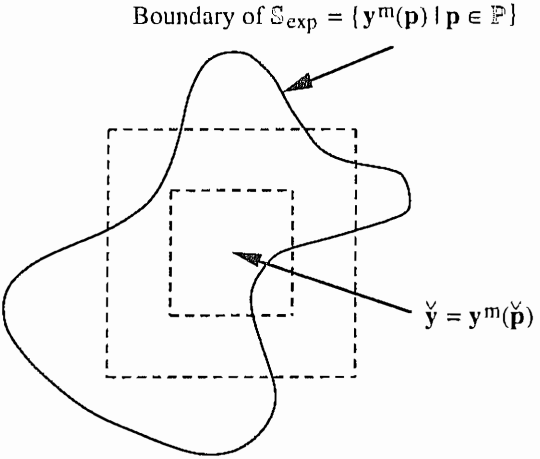

Boundary of $\mathbb{S}_{\mathrm{exp}} = \{\mathbf{y}^{\mathrm{m}}(\mathbf{p}) \mid \mathbf{p} \in \mathbb{P}\}$
$\check{\mathbf{y}} = \mathbf{y}^{\mathrm{m}}(\check{\mathbf{p}})$

Figure 4.41. Level sets of $\|\mathbf{y} - \mathbf{y}^{\mathrm{s}}\|_\infty$ on $\mathbb{H}_{\mathrm{exp}}$ (dashed lines) and boundary of $\mathbb{S}_{\mathrm{exp}}$ (solid line); the cost is inverse unimodal

By contrast, in Figure 4.42 the cost is inverse multimodal, because any $\mathbf{p}$ such that $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ is in the dashed region belongs to a basin of attraction that differs from that of $\check{\mathbf{p}}$. Indeed, $j(\mathbf{p})$ increases along any path contained in $\mathbb{S}_{\mathrm{exp}}$ and connecting $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ in the dashed area to $\check{\mathbf{y}}$.

The following two examples illustrate the case of least-squares estimation. We have seen in Section 3.1 (Remark 3.1) that replicated measurements can then be replaced by their mean (provided that the weights are suitably adjusted). When only n_{p} experimental conditions are used, it is therefore possible to work in an n_{p}-dimensional space, *i.e.* directly on $\mathbb{H}_{\mathrm{exp}}$.

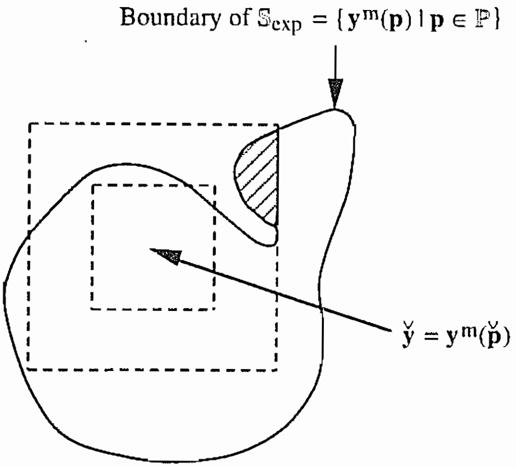

Boundary of $\mathbb{S}_{\rm exp} = \{\mathbf{y}^m(\mathbf{p}) \mid \mathbf{p} \in \mathbb{P}\}$
$\check{\mathbf{y}} = \mathbf{y}^m(\check{\mathbf{p}})$

Figure 4.42. Level sets of $\|\mathbf{y} - \mathbf{y}^s\|_\infty$ on $\mathbb{H}_{\rm exp}$ (dashed lines) and boundary of $\mathbb{S}_{\rm exp}$ (solid line); the cost is inverse multimodal

EXAMPLE 4.20

Consider the model response

$$y_m(t, p) = p^2 + p(1 - p)t.$$

Assume that two measurements are performed, at $t_1 = 0$ and $t_2 = 1$. When $\mathbf{y}^s = (2.5, 1)^T$, the least-squares criterion has a local optimum at $\hat{p}_{ls} = -1.2488$, and a global one at $\check{p}_{ls} = 1.5356$. Figure 4.43 shows the expectation surface $\mathbb{S}_{\rm exp}$ and the locations of the responses $\mathbf{y}^m(\hat{p}_{ls})$ and $\mathbf{y}^m(\check{p}_{ls})$. Assume now that the two observations $\mathbf{y}^s = (1.5, 1.1)^T$ are taken at the same time $t = 1$. The two components of $\mathbf{y}^m(p)$ then become identical, so all model responses are on the line bisecting the first quadrant of the observation space (Figure 4.44), and the local minimum disappears ($\hat{p}_{ls} = \check{p}_{ls} = 1.3$). ◊

EXAMPLE 4.21

Consider the model response described by

$$y_m(\xi, \mathbf{p}) = p_1\xi_1 + p_2\xi_2 + p_1^3(1 - \xi_1) + p_2^2(1 - \xi_2),$$

where the vector ξ characterizes the experimental conditions. Assume first that three observations are taken under the experimental conditions

$$\xi^1 = (1, 0)^T, \quad \xi^2 = (1, 1)^T \quad \text{and} \quad \xi^3 = (0, 1)^T.$$

Figure 4.45 presents the expectation surface $\mathbb{S}_{\rm exp}$ for $\mathbb{P} = [-5, 5] \times [-2, 5]$, and Figure 4.46 the corresponding least-squares cost contours when $\mathbf{y}^s = (5, -10, 8)^T$. One can easily check that the model parameters are globally identifiable under these experimental conditions (*i.e.* $\mathbf{y}^m(\mathbf{p}_1) = \mathbf{y}^m(\mathbf{p}_2) \Rightarrow \mathbf{p}_1 = \mathbf{p}_2$ almost everywhere), so that the inverse multimodality is not due to a lack of identifiability.

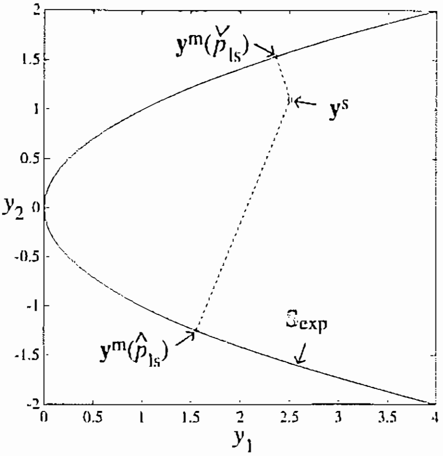


Figure 4.43. Expectation surface for Example 4.20; experimental conditions differ and there is a parasitic local minimum

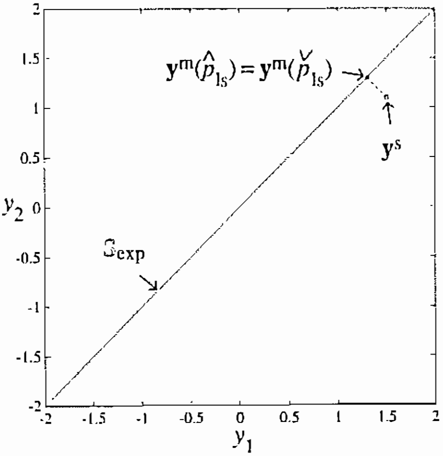


Figure 4.44. Expectation surface for Example 4.20; a single experiment is duplicated and the parasitic local minimum disappears

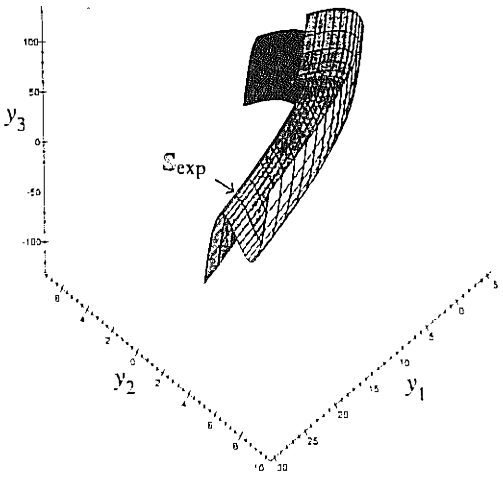


Figure 4.45. Expectation surface for Example 4.21 with three different experiments

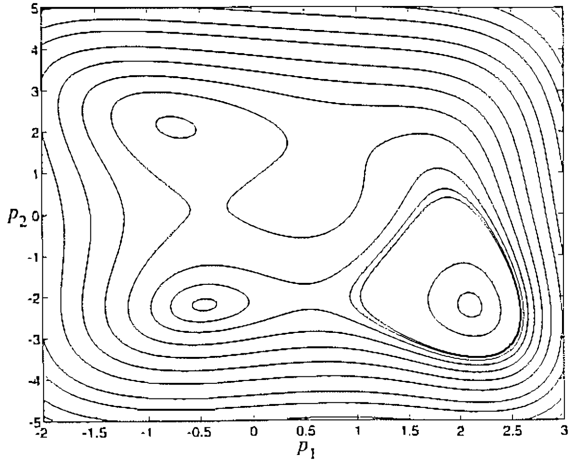


Figure 4.46. Cost contours for Example 4.21 with three different experiments; the cost is inverse multimodal, and two minimizers are local

Assume now that the experimental conditions are

$$\xi^1 = (1, 0)^{\mathrm{T}}, \quad \xi^2 = \xi^3 = (1, 1)^{\mathrm{T}}.$$

The expectation surface becomes flat, but the parameters then turn out to be only locally identifiable, because

$$\hat{\mathbf{p}}_1 = (\hat{p}_1, \hat{p}_2)^{\mathrm{T}} \quad \text{and} \quad \hat{\mathbf{p}}_2 = (\hat{p}_1 + 2\hat{p}_2 - 1, 1 - \hat{p}_2)^{\mathrm{T}}$$

give the same point $\mathbf{y}^m(\hat{\mathbf{p}}_1) = \mathbf{y}^m(\hat{\mathbf{p}}_2)$ on the expectation surface, and thus the same value of the cost. Figure 4.47 presents the least-squares cost contours when $\mathbf{y}^s = (5, 2, 4)^T$. The two global minimizers are $\hat{\mathbf{p}}_1 = (1, 2)^T$ and $\hat{\mathbf{p}}_2 = (4, -1)^T$. Using a local optimization method of Section 4.3.3, either of them can be obtained, from which the other is easily calculated. Compare with Example 4.18, where the same model structure was considered in the context of least modulus. ◊

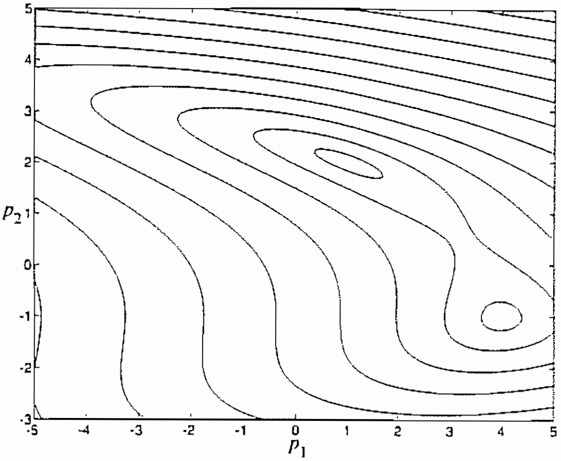


Figure 4.47. Cost contours for Example 4.21 with repetition of only two different experiments; the cost remains inverse multimodal, but both minimizers are now global

4.3.9.2 Random search

A basic algorithm for random-search minimization is as follows:

Step 1: Choose $\hat{\mathbf{p}}^0$, set $k = 0$.
Step 2: Compute a trial point $\hat{\mathbf{p}}^{k+} \in \mathbb{P}$ according to the rule

$$\hat{\mathbf{p}}^{k+} = \hat{\mathbf{p}}^k + \mathbf{r}^k,$$

where $\mathbf{r}^k$ is a realization of a suitably distributed random vector.
Step 3: If $j(\hat{\mathbf{p}}^{k+}) < j(\hat{\mathbf{p}}^k)$ then $\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^{k+}$, else $\hat{\mathbf{p}}^{k+1} = \hat{\mathbf{p}}^k$.
Step 4: Increment k by one and go to Step 2.

Let $\check{\mathbf{p}}$ be a global minimizer to be located. When $\hat{\mathbf{p}}^k$ is far from $\check{\mathbf{p}}$, $\mathbf{r}^k$ should have a large variance to allow large displacements that might be necessary to escape the attraction of local minimizers. Conversely, when $\hat{\mathbf{p}}^k$ is near $\check{\mathbf{p}}$, $\mathbf{r}^k$ should have a small variance to allow finer exploration of parameter space. The idea of adaptive random search (Bekey and Masri, 1983; Pronzato *et al.*, 1984) is to alternate *variance-selection phases* and *variance-exploitation phases*, during which the selected variance is used.

In the version that we have implemented, the user must only provide

— the prior feasible domain for the parameters, assumed to be the box

$$\mathbb{P} = \{\mathbf{p} \mid p_{i_{\min}} \leq p_i \leq p_{i_{\max}}, i = 1, \dots, n_{\mathrm{p}}\},$$

— a subroutine computing $j(\mathbf{p})$ for any $\mathbf{p}$ in $\mathbb{P}$,
— the maximum number of evaluations of the cost allowed.

Unless otherwise specified, search is initialized at the centre of $\mathbb{P}$

$$\hat{p}_i^0 = \frac{p_{i_{\min}} + p_{i_{\max}}}{2}, \quad i = 1, \dots, n_{\mathrm{p}}.$$

The displacement $\mathbf{r}^k$ is randomly generated according to an $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}(\boldsymbol{\sigma}))$ distribution with

$$\mathbf{\Sigma}(\boldsymbol{\sigma}) = \operatorname{diag}\{\sigma_i^2, i = 1, \dots, n_{\mathrm{p}}\},$$

truncated so that $\hat{\mathbf{p}}^{k+}$ belongs to $\mathbb{P}$. (A uniform distribution could also be used.)

Variance-selection phase. Several successive values of $\boldsymbol{\sigma}$ are tried for a given number of iterations of the basic algorithm. One may, for instance, choose

$$^1\boldsymbol{\sigma} = \mathbf{p}_{\max} - \mathbf{p}_{\min},$$

which promotes large displacements in $\mathbb{P}$, and

$$^i\boldsymbol{\sigma} = {}^{i-1}\boldsymbol{\sigma}/10, \; i = 2, \dots, 5,$$

for finer and finer explorations. The competing $^i\boldsymbol{\sigma}$'s are rated by their performance in the basic algorithm in terms of cost reduction *starting from the same initial point*, namely the best $\hat{\mathbf{p}}^k$ available at the start of the comparison. Each $^i\boldsymbol{\sigma}$ may for example be allowed $100/i$ iterations to give more trials to larger variances. The best $^i\boldsymbol{\sigma}$ in terms of the final value of the cost, denoted by $\hat{\boldsymbol{\sigma}}$, is selected for the variance-exploitation phase.

Variance-exploitation phase. Starting from the best $\hat{\mathbf{p}}^k$ obtained during the variance-selection phase, the basic algorithm is used with the covariance $\mathbf{\Sigma}(\hat{\boldsymbol{\sigma}})$ for, typically, one hundred iterations before resuming a variance-selection phase.

Local optimization. Since the smallest $\boldsymbol{\sigma}$ corresponds to very small displacements in parameter space, one may switch to local optimization whenever $^5\boldsymbol{\sigma}$ is selected (Section 4.3.3). In order to avoid uselessly duplicating local optimizations, these will be performed only if $^5\boldsymbol{\sigma}$ is selected for the first time or if $^5\boldsymbol{\sigma}$ was not the previous $\hat{\boldsymbol{\sigma}}$.

Termination. The algorithm is terminated when the maximum number of cost evaluations allowed is reached, or when $^5\boldsymbol{\sigma}$ has been selected a given number of times consecutively, which indicates that the algorithm has failed to escape the basin of attraction of the best local minimizer so far.

PARS1: Except for pathological cost functions such that any global minimizer has a basin of attraction with zero measure, if the number of evaluations of the cost tends to infinity, $\hat{\mathbf{p}}^k$ will tend to a global minimizer of j over $\mathbb{P}$. Of course, this does not guarantee that any global minimizer will be reached, since the actual number of iterations is always finite.

PARS2: This very simple method is not aimed at finding all global minimizers, but merely one of them.

PARS3: The treatment of a dozen test problems from the literature (Pronzato *et al.*, 1984) has shown that adaptive random search located global optimizers at least as efficiently as the global optimization methods advocated in the papers describing these test problems. Although limited in scope, this comparison indicates that adaptive random search can handle varied problems with several local minimizers, without any modification to the algorithm or adaptation of its internal parameters.

PARS4: The time spent selecting the variance is usually profitable. (See Example 4.22.)

PARS5: Provided that the cost function is twice continuously differentiable, switching to a local method when ${}^5\boldsymbol{\sigma}$ is selected in general much improves the performance.

PARS6: The method applies to non-differentiable and discontinuous cost functions, such as the number of sign changes mentioned in Section 3.7.4 or the percentage of data considered as outliers to be presented in Section 5.4.2.2.

EXAMPLE 4.22

Consider the experiment design problem (Chapter 6) defined by Bohachevsky, Johnson and Stein (1986). The system studied is described by

$$y(t_i, t_{i-1}) = y_{\mathrm{m}}(t_i, t_{i-1}, \mathbf{p}) + \varepsilon(t_i),\ i = 1, \ldots, 11,$$

with $\varepsilon(t_i)$ belonging to a sequence of i.i.d. random variables,

$$y_{\mathrm{m}}(t_i, t_{i-1}, \mathbf{p}) = p_1[\exp(-p_3 t_{i-1}) - \exp(-p_3 t_i)] + p_2(t_i - t_{i-1}),$$

and $t_0 = 0$. The vector $\mathbf{t} = (t_1, \ldots, t_{11})^{\mathrm{T}}$ should be chosen so as to optimize a cost quantifying the precision with which the parameters $\mathbf{p}$ will be estimated. The criterion chosen is D-optimality (Section 6.1), which corresponds to maximizing

$$j(\mathbf{t}) = \det\left[\frac{\partial \mathbf{y}^{\mathrm{m}\mathrm{T}}(\mathbf{p})}{\partial \mathbf{p}} \frac{\partial \mathbf{y}^{\mathrm{m}}(\mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}\right].$$

The prior feasible set $\mathbb{T}$ for $\mathbf{t}$ is defined by

$$\mathbb{T} = \{\mathbf{t} \in \mathbb{R}^{11} \mid t_1 \geq 1,\ t_i - t_{i-1} \geq 1,\ \ i = 2, \ldots, 11,\ \ \mathrm{t}_{11} \leq 30\ \},$$

and p_3 is taken to be 0.25. (The value of $\mathbf{t}$ that maximizes $j(\mathbf{t})$ does not depend on p_1 and p_2.) After about 5000 evaluations of the cost, adaptive random search implemented as above (with projection onto $\mathbb{T}$ of any trial point $\hat{\mathbf{t}}^{k+}$ violating the constraints) suggests

$$\hat{\mathbf{t}} \approx (3.2,\ 11.2,\ 12.2,\ 13.2,\ 14.2,\ 15.2,\ 16.2,\ 17.2,\ 18.2,\ 19.2,\ 30)^{\mathrm{T}},$$

i.e. the solution obtained by Bohachevsky, Johnson and Stein (1986) using simulated annealing, after several thousand evaluations of the cost. Figure 4.48 shows a typical evolution of $j(\hat{\mathbf{t}}^{k+})$ with k. The alternating variance-selection and exploitation phases are easily recognized.

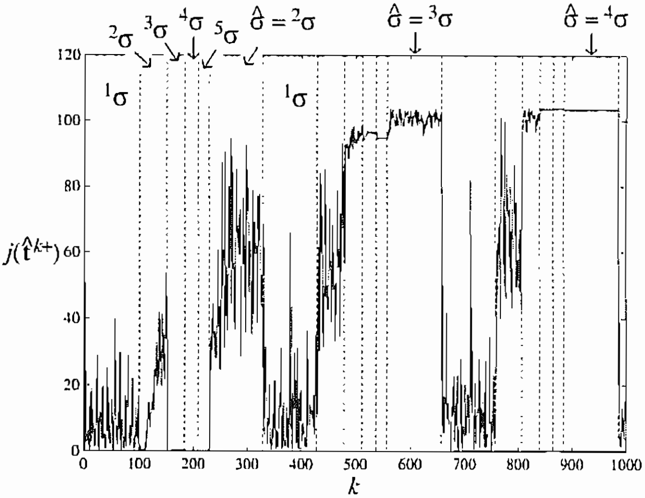


Figure 4.48. Typical evolution of $j(\hat{\mathbf{t}}^{k+})$ with k

For comparison, Figure 4.49 presents a typical evolution of $j(\hat{\mathbf{t}}^{k+})$ when the basic algorithm is used with $\boldsymbol{\sigma}$ cyclically taking the values ${}^1\boldsymbol{\sigma}$, ${}^2\boldsymbol{\sigma}$, ... , ${}^5\boldsymbol{\sigma}$, each for one hundred iterations, instead of alternating variance-selection and exploitation phases. The result is not nearly as good as with adaptive random search. Restarting at the same $\hat{\mathbf{t}}^k$ to compare the various ${}^i\boldsymbol{\sigma}$'s during the variance-selection phase penalizes adaptive random search, resulting in a quicker initial increase of $j(\hat{\mathbf{t}}^{k+})$ in Figure 4.49 than in Figure 4.48. This, however, is more than cancelled by the gain in efficiency resulting from the use of a suitable variance (here ${}^3\boldsymbol{\sigma}$) for a large number of iterations. ◊

4.3.9.3 Deterministic search

Many experts in optimization thought (some even wrote) that it was impossible to develop techniques guaranteed to deliver all global optimizers of a multimodal cost function. *Interval analysis*, a very active research field for the past twenty years (Adams and Kulisch, 1993; Kearfott and Kreinovich, 1996, Moore, 1979), provided a spectacular refutation, even if the techniques derived from this approach are limited in the complexity of the problems that they can solve.

Interval analysis. An *interval* $[x]$ of $\mathbb{R}$ (or *scalar interval*) is a connected closed and bounded set of real numbers

$$[x] = [x^-, x^+] = \{x \mid x^- \leq x \leq x^+\}.$$

Interval arithmetic extends computation on real numbers to intervals in a natural and intuitive way. The product of an interval by a real scalar λ is, for instance, given by

$$\text{if } \lambda \geq 0, \text{ then } \lambda[x] = [\lambda x^-, \lambda x^+], \text{ else } \lambda[x] = [\lambda x^+, \lambda x^-].$$

Similarly

$$[x] + [y] = [x^- + y^-, x^+ + y^+],$$

$$[x] - [y] = [x^- - y^+, x^+ - y^-],$$

$$[x][y] = [\min(x^-y^-, x^-y^+, x^+y^-, x^+y^+), \max\,(x^-y^-, x^-y^+, x^+y^-, x^+y^+)],$$

$$1/[x] = [1/x^+, 1/x^-], \text{ provided that } 0 \notin [x],$$

$$\exp([x]) = [\exp(x^-), \exp(x^+)],$$

and simple algorithms can be provided to compute sin([x]), cos([x])... To allow division by intervals containing zero, extended intervals with possibly infinite bounds must be considered.

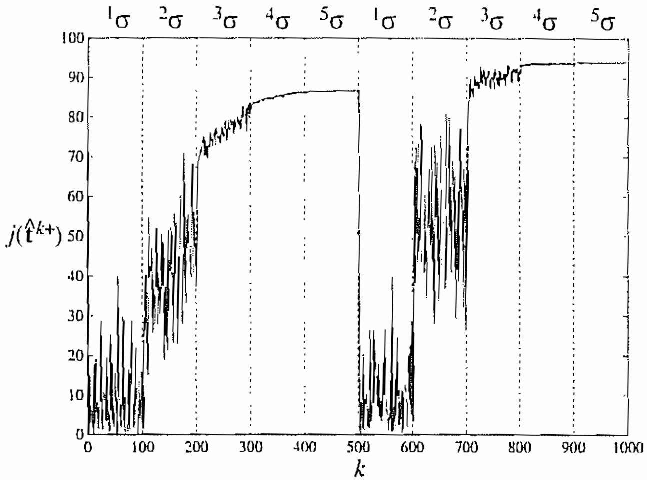


Figure 4.49. Typical evolution of $j(\hat{t}^{k+})$ for a cyclic change of the variances

It is important to note that the intervals [x] and [y] have implicitly been assumed to be independent. If, for example, it is known that $x = y$, then using the above formulas to evaluate $[x] - [y]$ or $[x][y]$ may yield a very pessimistic interval, nevertheless guaranteed to contain the actual set of possible values for the result. The square of [x] should therefore not be computed as $[x][x]$, but as follows:

$$\text{if } 0 < x^-, \text{ then } [x]^2 = [(x^-)^2, (x^+)^2],$$
$$\text{if } x^+ < 0, \text{ then } [x]^2 = [(x^+)^2, (x^-)^2],$$
$$\text{if } 0 \in [x], \text{ then } [x]^2 = [0, (\max(-x^-, x^+))^2].$$

Extensions of FORTRAN (IBM, 1986), C (Klatte *et al.*, 1993) and PASCAL (Klatte *et al.*, 1992) handle computation on intervals as well as on reals. A shareware version of the C-XSC library is available in several machine-readable formats (Hammer *et al.*, 1995). To take the errors due to finite word length into account, *outward rounding* is performed, which makes it possible to produce intervals guaranteed to contain the exact results. The lengths of these intervals indicate the uncertainty in these results, possibly very pessimistically.

A *box* $[\mathbf{p}]$ in $\mathbb{R}^{n_\mathrm{p}}$ (or *vector interval*) is the Cartesian product of n_p scalar intervals, which will be indifferently denoted by

$$[\mathbf{p}] = [p_1^-, p_1^+] \times [p_2^-, p_2^+] \times \ldots \times [p_{n_\mathrm{p}}^-, p_{n_\mathrm{p}}^+] = [p_1] \times [p_2] \times \ldots \times [p_{n_\mathrm{p}}] = [\mathbf{p}^-, \mathbf{p}^+].$$

Any box is therefore characterized by two extreme vectors $\mathbf{p}^-$ and $\mathbf{p}^+$. Any vector $\mathbf{p}$ can also be considered as a box, such that $\mathbf{p}^+ = \mathbf{p}^- = \mathbf{p}$. The set of all boxes of $\mathbb{R}^{n_\mathrm{p}}$ will be denoted by $\mathbb{IR}^{n_\mathrm{p}}$.

When boxes are used instead of vectors, point values of $\mathbf{p}$ are replaced by uncountable subsets of parameter space. This makes it possible to perform a *global* analysis with a *finite* number of operations.

Extending functions to the interval type allows concepts of vector arithmetic to be extended to boxes. Ideally, for any function $\mathbf{f}: \mathbb{R}^{n_\mathrm{p}} \to \mathbb{R}^{n_\mathrm{f}}$ and any box $[\mathbf{p}]$ of $\mathbb{IR}^{n_\mathrm{p}}$, one would like to get a function $[\mathbf{f}]: \mathbb{IR}^{n_\mathrm{p}} \to \mathbb{IR}^{n_\mathrm{f}}$ that would compute the smallest box of $\mathbb{IR}^{n_\mathrm{f}}$ that contains the set $\mathbf{f}([\mathbf{p}])$. Unfortunately, this is only possible for elementary functions such as sin or cos. When $[\mathbf{f}]([\mathbf{p}])$ cannot be evaluated, it can usually be approximated by using an *inclusion function* $\mathbb{f}: \mathbb{IR}^{n_\mathrm{p}} \to \mathbb{IR}^{n_\mathrm{f}}$, *i.e.* a function $\mathbb{f}$ that satisfies

$$\forall\ [\mathbf{p}] \in \mathbb{IR}^{n_\mathrm{p}},\ \mathbf{f}([\mathbf{p}]) \subset \mathbb{f}([\mathbf{p}]) \in \mathbb{IR}^{n_\mathrm{f}}.$$

As illustrated by Figure 4.50, $\mathbf{f}([\mathbf{p}]) \subset [\mathbf{f}]([\mathbf{p}]) \subset \mathbb{f}([\mathbf{p}])$. The inclusion function $\mathbb{f}$ therefore makes it possible to approximate the set $\mathbf{f}([\mathbf{p}])$ we are interested in, but cannot usually compute, by a computable box $\mathbb{f}([\mathbf{p}])$ guaranteed to contain it.

This box may however be too pessimistic to be of any practical use. One would therefore like the option of getting better approximations by considering smaller boxes of $\mathbb{IR}^{n_\mathrm{p}}$, for instance by splitting large boxes into sub-boxes. This is why the following two properties are highly desirable. An inclusion function $\mathbb{f}: \mathbb{IR}^{n_\mathrm{p}} \to \mathbb{IR}^{n_\mathrm{f}}$ is *inclusion monotonic* if

$$\forall\ [\mathbf{p}_1], [\mathbf{p}_2] \in \mathbb{IR}^{n_\mathrm{p}},\ [\mathbf{p}_1] \subset [\mathbf{p}_2] \Rightarrow \mathbb{f}([\mathbf{p}_1]) \subset \mathbb{f}([\mathbf{p}_2]).$$

It is *convergent* if

$$\forall\ [\mathbf{p}],\ w([\mathbf{p}]) \to 0 \Rightarrow w(\mathbb{f}([\mathbf{p}])) \to 0,$$

where the *width* $w([\mathbf{p}])$ of a box $[\mathbf{p}]$ is defined as the length of its largest side(s). When the box $[\mathbf{p}]$ tends to a vector $\mathbf{p}$, its image by a convergent inclusion function therefore tends to the vector $\mathbf{f}(\mathbf{p})$. If the effect of rounding is neglected, it is very simple to derive an inclusion-monotonic and convergent inclusion function $\mathbb{f}$ for any continuous function $\mathbf{f}$ defined by an explicit formal expression (or program). It suffices to replace all elementary operators and functions such as +, –, $*$, /, sin, cos, exp… by their interval counterpart as defined above. Note that there are infinitely many inclusion functions associated with a given function $\mathbf{f}$, so accuracy might be improved by intersecting all the boxes $\mathbb{f}_i([\mathbf{p}])$ associated with various inclusion functions $\mathbb{f}_i$.

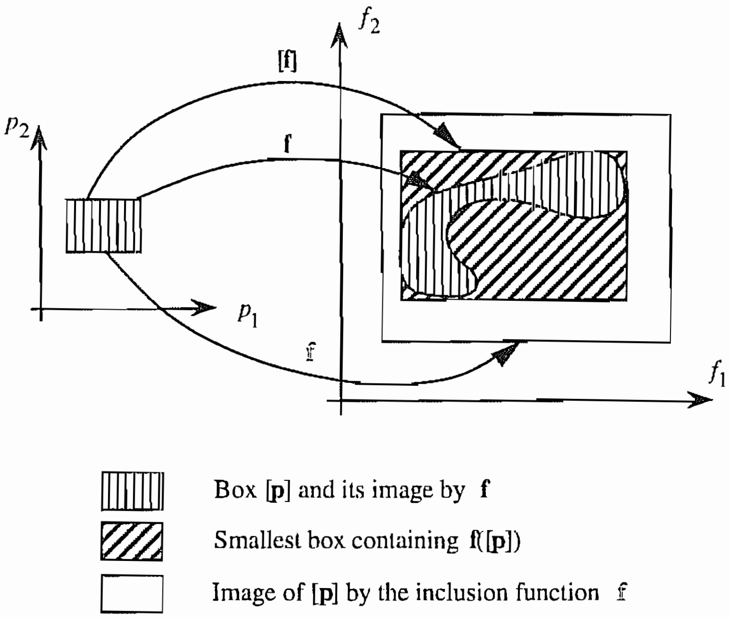

f_2
[f]
p_2
f
p_1
$\mathbb{f}$
f_1
▥ Box [p] and its image by f
▨ Smallest box containing f([p])
□ Image of [p] by the inclusion function $\mathbb{f}$

Figure 4.50. Inclusion functions

EXAMPLE 4.23

Consider the function $f\colon \mathbb{R} \to \mathbb{R}$ defined by $f(x) = x^2 - 2x + 1$. A first possible inclusion function for f is defined by $\mathbb{f}_1([x]) = [x]^2 - 2[x] + 1$. We have, for instance,

$$\mathbb{f}_1([3, 4]) = [3, 4]^2 - 2[3, 4] + [1, 1] = [9, 16] + [-8, -6] + [1, 1] = [2, 11].$$

In performing these computations, we have neglected the fact that $[x]^2$ and $2[x]$ are obviously not independent intervals. As a result, $\mathbb{f}_1$ yields a pessimistic interval. To get more accurate results, one should try to minimize the number of occurrences of each variable in the formal definition of the function. In this example, it is possible for x to appear only once, since $f(x) = (x - 1)^2$. A much better inclusion function will therefore be defined by $\mathbb{f}_2([x]) = ([x] - 1)^2$, such that

$$\mathbb{f}_2([3,4]) = ([3,4]-[1,1])^2 = [2,3]^2 = [4,9].$$

The result is no longer pessimistic, and $\mathbb{f}_2([x]) = f([x])$. More generally, if $\mathbf{f}$ is a vector function, exact intervals will be obtained for each component of $\mathbf{f}([\mathbf{x}])$ provided that each interval variable appears at most once in the formal expression of each component of $\mathbb{f}$.

Assume now that $f(2)$ has been evaluated, $f(2) = 1$. Since $f(2)$ is smaller than the lower bound for $f([3,4])$ as obtained from $\mathbb{f}_2([3,4])$, we have *numerically proved* that the interval [3, 4] contains no unconstrained global minimizer of f. This possibility of eliminating subsets of parameter space that cannot contain optimizers is at the core of the algorithm to be described. ◊

As illustrated by the previous example, inclusion functions generated by automated replacement of all elementary operators and functions by their interval counterparts are usually very pessimistic, because the relationships existing between the results of intermediary computations are neglected, which results in the so-called *wrapping effect*. The approximation can be improved by splitting boxes into sub-boxes to take advantage of inclusion monotonicity and convergence, but it is clear that using good inclusion functions can tremendously improve the performance of the algorithm presented next.

Algorithm. This algorithm is a simplified version of that described in (Hansen, 1992), which we consider mandatory reading for anyone interested. To present it, we shall assume that the cost function j is twice continuously differentiable over the whole prior feasible domain. We shall moreover assume that the global minimum of the cost corresponds to a stationary point (unconstrained optimization).

Let

— $\mathbb{L}_\mathrm{i}$ be the list of all boxes still to be treated by the algorithm,
— $\mathbb{L}_\mathrm{s}$ be the list of all boxes (and associated characteristics) that may contain feasible global minimizers of the cost upon completion of the algorithm,
— $\mathbf{c}([\mathbf{p}])$ be the centre of the box $[\mathbf{p}]$,
— $\overline{j}$ be an upper bound for the global minimum $\check{j}$,
— $\underline{j}$ be a lower bound for $\check{j}$.

The user must provide

— an initial list $\mathbb{L}_\mathrm{i}$, specifying the box or set of (possibly disconnected) boxes within which the search is to be performed,
— the rule for evaluating the cost function j and an associated inclusion function $\mathbb{j}$,
— the accuracy δ_j required for the determination of $\check{j}$.

If possible, the user should also provide

— an inclusion function $\mathbb{g}$ for the gradient $\mathbf{g}$ of the cost function,
— inclusion functions $\mathbb{h}_{ii}$ for the diagonal entries h_{ii} of the Hessian $\mathbf{H}$ of the cost function $(i = 1, \ldots, n_\mathrm{p})$.

Else, the steps that become impossible (Step 4 or 5) will be omitted, at the cost of decreased efficiency.

The lists $\mathbb{L}_\mathrm{i}$ and $\mathbb{L}_\mathrm{s}$ will contain the location of each box $[\mathbf{p}]$ that belongs to them (*i.e.* $\mathbf{p}^-$ and $\mathbf{p}^+$) and, as soon as computed, a lower bound j^- for the value of the cost function over $[\mathbf{p}]$.

Step 1: Evaluate an upper bound for $\check{j}$ as

$$\overline{j} = \min_{[\mathbf{p}]\in\mathbb{L}_\mathrm{i}} j(\mathbf{c}([\mathbf{p}])).$$

Step 2: For each box $[\mathbf{p}]$ in $\mathbb{L}_\mathrm{i}$, compute

$$[j^-, j^+] = \mathbb{j}([\mathbf{p}]).$$

If $j^- > \overline{j}$, discard $[\mathbf{p}]$ from $\mathbb{L}_\mathrm{i}$ (for it contains no global minimizer $\check{\mathbf{p}}$). Else, store j^- as one of the characteristics of $[\mathbf{p}]$ that can be retrieved from $\mathbb{L}_\mathrm{i}$.

Step 3: If $\mathbb{L}_\mathrm{i}$ is empty (which cannot happen during the first iteration), go to Step 8. Else, select a box $[\mathbf{p}]$ of $\mathbb{L}_\mathrm{i}$ associated with the smallest value of j^-.

Step 4: If $\mathbf{0} \notin \mathbb{g}([\mathbf{p}])$, discard $[\mathbf{p}]$ from $\mathbb{L}_\mathrm{i}$ (for it contains no stationary point of j), and go to Step 3.

Step 5: If there exists i such that $\mathbb{h}_{ii}([\mathbf{p}]) < 0$, discard $[\mathbf{p}]$ from $\mathbb{L}_\mathrm{i}$ (for j is not convex in the neighbourhood of any stationary point in $[\mathbf{p}]$, which therefore contains no unconstrained minimizer), and go to Step 3.

Step 6: Remove $[\mathbf{p}]$ from $\mathbb{L}_\mathrm{i}$ and split it into 2^k boxes by k bisections perpendicular to the axes of parameter space along which the length of $[\mathbf{p}]$ is largest. Typically, $k = \min(3, \dim \mathbf{p})$.

Step 7: For each of the boxes $[\mathbf{p}]$ created at Step 6,

```
  {update $\overline{j}$ as $\overline{j} = \min(\overline{j}, j(\mathbf{c}([\mathbf{p}])))$;
  compute $[j^-, j^+] = \mathbb{j}([\mathbf{p}])$;
  if $j^- > \overline{j}$, discard $[\mathbf{p}]$; else
      {store $j^-$ as one of the characteristics of $[\mathbf{p}]$;
      if $w(\mathbb{j}([\mathbf{p}])) < \delta_\mathrm{j}$, store $[\mathbf{p}]$ in $\mathbb{L}_\mathrm{s}$; else store $[\mathbf{p}]$ in $\mathbb{L}_\mathrm{i}$}}.
```

When the 2^k boxes created at Step 6 have been processed, go to Step 3.

Step 8: $\mathbb{L}_\mathrm{i}$ is empty. Discard from $\mathbb{L}_\mathrm{s}$ any $[\mathbf{p}]$ such that $j^-([\mathbf{p}]) > \overline{j}$. Stop.

The remaining boxes in $\mathbb{L}_\mathrm{s}$ are the only ones that may contain a global minimizer. All of them satisfy

$$j([\mathbf{p}]) - \check{j} < \delta_\mathrm{j}.$$

A lower bound for the global minimum $\check{j}$ is given by

$$\underline{j} = \min_{[\mathbf{p}]\in\mathbb{L}_\mathrm{s}} j^-([\mathbf{p}]).$$

Upon completion of the algorithm, $\check{j}$ is known to belong to $[\underline{j}, \overline{j}]$ and all unconstrained global minimizers in the domain of interest are known to belong to the union of all boxes of $\mathbb{L}_\mathrm{s}$.

REMARKS 4.26

— The policy used for selecting [**p**] at Step 3 is particularly efficient, and constitutes an essential ingredient of the algorithm.

— A necessary condition for a stationary point **p** to be an unconstrained minimizer is that **H**(**p**) be nonnegative definite, *i.e.* that all its eigenvalues be positive or zero. A simpler-to-test necessary condition is that all diagonal entries of **H**(**p**) be positive or zero. This is the condition exploited at Step 5. More complicated tests could be considered, but a compromise must be struck between complexity and added efficiency in terms of box elimination.

— If the list $\mathbb{L}_s$ turns out to be empty on completion of the algorithm, this has only two possible explanations. The first is that the global minimum of the cost function j in the boxes of the initial list $\mathbb{L}_i$ does not correspond to a stationary point, which amounts to saying that the global optimum is reached on the border of the prior feasible domain. One should then either enlarge this prior domain or turn to constrained optimization. Variants are available to deal with equality or inequality constraints (Hansen, 1992). The second explanation would be that an error has been committed when implementing the algorithm, *e.g.* in specifying the inclusion functions!

— We have assumed here that it is always possible to improve the accuracy of the results provided by the inclusion functions by reducing the size of the boxes considered. In practice, to get guaranteed results one must take into account the numerical errors induced by finite word length in the machine representation of numbers. Inclusion functions must therefore be evaluated with outward rounding. As a result, the inclusion functions are no longer convergent, the image of a vector being already a box with nonzero width. This limits the accuracy that can be achieved. This is why the bisections at Step 6 will only be performed if the width of the box [**p**] to be bisected is larger than a given threshold δ_p. Else, [**p**] will be put in $\mathbb{L}_s$ without attempting to analyse it in any more detail.

— Exact methods for evaluating the gradient **g** (and possibly the Hessian **H**) of the cost function will form the basis for working out the corresponding inclusion functions.

— Whenever the algorithm decreases $\bar{j}$, one might try to improve the resulting bound further by starting a local optimization from the corresponding value of **c**([**p**]).

— A better bisection policy at Step 6 is to split [**p**] perpendicular to axes i of parameter space associated with the largest values of $d_i = w(g_i([\mathbf{p}]))w([\mathbf{p}]_i)$, where $g_i([\mathbf{p}])$ and $[\mathbf{p}]_i$ are the scalar intervals corresponding to the ith components of **g**([**p**]) and [**p**]. It is thus possible to take into account the speed of variation of the cost in the various possible directions.

— The smaller δ_j is, the more intensive the computation becomes. One should therefore limit the required precision to what is necessary. So long as the representation of numbers in the computer is precise enough, one might always restart from a result that is deemed too coarse and make it more accurate by reducing δ_j and taking the list $\mathbb{L}_s$ just obtained as the new initial list $\mathbb{L}_i$. ◊

A description of the performance of a variant of this algorithm will be found in (Hansen, 1992). The test problems treated have up to 50 parameters, up to 10^{10} local minimizers and up to a continuum of global minimizers. The variant used includes an interval Newton method for solving $\mathbf{g}(\mathbf{p}) = \mathbf{0}$, which makes it possible to locate the stationary points within the boxes considered much more rapidly.

4.4 Optimization of a measured response

The problem considered now is optimization of the response y of a system with respect to n_f independent *factors*, or *operating conditions*. The vector ξ of these operating conditions must belong to some prior feasible space Ξ. This problem, of obvious practical importance (*e.g.* in quality control), has received considerable attention in the literature. Its solution is tricky, because the exact dependence of y on ξ is usually unknown, and because y is measured and therefore subject to random variability. Knowledge about the process is deduced from measurements under operating conditions ξ^i ($i = 1, \ldots, n_t$). This knowledge will be used to predict the deterministic part of $y(\xi)$, denoted by $y_m(\xi)$, then to infer the value ξ^* that optimizes (maximizes in what follows) $y_m(\xi)$. In this context, an efficient method is therefore one that yields an accurate estimate of ξ^* from few measurements $y(\xi^i)$. The available methods differ mainly in their choice of the ξ^i's (experiment design, see Chapter 6) and the estimator used for ξ^*. The model $y_m(\xi)$ is merely a step in the computation of ξ^*, and some methods, as we shall see in the next section, do not even use an explicit model. When such a model is employed, it is usually a simple LP behavioural model, and the issue of robustness of the solution with regard to model structure is therefore important (see Section 6.6.2 for a brief survey of the methods proposed in the literature).

4.4.1 Model-free optimization

The most intuitive approach is to use a derivative-free optimization method, such as the *simplex* algorithm (Nelder and Mead, 1965); see Remark 4.13. Each evaluation of the cost is performed by a measure on the process, and at each step, given the past observations, new operating conditions are suggested. This approach is therefore model-free, and interesting when nothing is known about the dependence of y on ξ. Should the deterministic part $y_m(\xi)$ of the response be accessible to measurement, the simplex method would generally lead to a local maximum of y_m. The variability of $y(\xi)$ due to noise is, however, not taken into account, so the suggested operating conditions do not converge to ξ^* but fluctuate randomly.

The random variability of $y(\xi)$ can be taken into account with the help of *stochastic approximation* (Kiefer and Wolfowitz, 1952). A variant of the stochastic gradient algorithm presented in Section 4.3.8 can be used when the gradient of the cost function is not measurable:

$$\xi_n^{i+1} = \xi_n^i + \frac{\gamma_i}{\alpha_i}\,[y(\xi^i + \alpha_i \mathbf{e}^n) - y(\xi^i)], \quad n = 1, \ldots, n_f,$$

where ξ_n^i is the nth component of ξ^i and $\mathbf{e}^n$ is the unit vector giving axis n in $\mathbb{R}^{n_f}$. The scalars α_i and γ_i must satisfy

$$\lim_{i\to\infty} \gamma_i = \lim_{i\to\infty} \alpha_i = 0, \quad \sum_{i=1}^{\infty} \gamma_i = \infty, \quad \sum_{i=1}^{\infty} \left(\frac{\gamma_i}{\alpha_i}\right)^2 < \infty.$$

The directions in which the operating conditions are varied again become more and more arbitrary when the search gets closer to $\boldsymbol{\xi}^*$, but the decrease of the step length ensures convergence to a local maximum (Dvoretzky, 1956; Polyak and Tsypkin, 1973; Saridis, 1974). The method is quite general and simple (no model structure need be assumed), but the number of measurements required is usually high (as convergence is slow and each iteration requires $n_{\mathrm{f}} + 1$ measurements), much larger than with a model-based approach such as described in the next section. Accelerated variants are presented in (Spall, 1992, 1995).

4.4.2 Response-surface methodology

Response-surface methodology is a set of mathematical and statistical tools aimed at locating $\boldsymbol{\xi}^*$ (Box and Wilson, 1951; Hill and Hunter, 1966; Mead and Pike, 1975; Myers, 1976; Box and Draper, 1987), and based upon sequential construction of a suitable measurement scheme. At each step, a small domain ξ centred on the current estimate of $\boldsymbol{\xi}^*$ is considered.

The *response surface* (not to be confused with the expectation surface $\mathbb{S}_{\mathrm{exp}}$ of Sections 4.3.9.1 and 5.1.1.1) is defined as

$$\mathbb{S}_{\mathrm{resp}} = \left\{ \begin{bmatrix} \boldsymbol{\xi} \\ y_{\mathrm{m}}(\boldsymbol{\xi}) \end{bmatrix}, \ \boldsymbol{\xi} \in \xi \right\}.$$

During the initial steps, when $\boldsymbol{\xi}$ is far from the optimum, the curvature of $\mathbb{S}_{\mathrm{resp}}$ can be neglected, and a linear function of $\boldsymbol{\xi}$ can be used (first-degree model):

$$y_{\mathrm{m}}(\boldsymbol{\xi}) = q_0 + \sum_{n=1}^{n_{\mathrm{f}}} q_n \xi_n .$$

Since the objective is to reach $\boldsymbol{\xi}^*$ as quickly as possible, a natural approach is steepest ascent (Section 4.3.3.1), based on an evaluation of the gradient of $y_{\mathrm{m}}(\boldsymbol{\xi})$, *i.e.* of q_n ($n = 1, \ldots, n_{\mathrm{f}}$) (Montgomery, 1976). The step sizes are fixed *a priori*, and steps are performed in the same estimated gradient direction as long as y_{m} keeps increasing. A new region is then considered to compute a new estimate of the gradient. The choice of the $\boldsymbol{\xi}^i$'s for this estimation is a classical experiment-design problem, and a first-order orthogonal array can be used (Box and Wilson, 1951). The choice of levels of the operating conditions to be used, which specifies the size of the region ξ, is considered by Steinberg (1985) following a Bayesian approach, with robustness with respect to the model structure in mind. The accuracy with which the gradient, and therefore the direction to be followed, are estimated increases with the number of measurements. A compromise must therefore be struck between precision and cost. For a fixed-length sequence of steps, Brooks and Mickey (1961) show that the ratio of the increase in the response to the number of measurements is a maximum when the number of measurements used to estimate the gradient is a minimum, $n_{\mathrm{f}} + 1$. The stochastic-approximation sequence of the previous section estimates the gradient with a design defined by $\{\mathbf{0}, \mathbf{e}^1, \ldots, \mathbf{e}^{n_{\mathrm{f}}}\}$. Other designs with $n_{\mathrm{f}} + 1$ support points, such as simplex designs, are more suitable, however. They should not be confused with the simplex designs used to choose the optimal composition of a product, where the factors must satisfy

$$\xi_n \geq 0, \quad \sum_{n=1}^{n_f} \xi_n = 1$$

(Scheffé, 1958; Kiefer, 1961; Galil and Kiefer, 1977).

When the estimate of $\boldsymbol{\xi}^*$ gets closer to its actual value, the curvature of $\mathbb{S}_{\text{resp}}$ can no longer be neglected, and the degree of the model should (at least) be two. (See (Montgomery, 1976) for tests on the validity of first-degree models.) Most often, the model used is quadratic in $\boldsymbol{\xi}$:

$$y_{\text{m}}(\boldsymbol{\xi}) = q_0 + \sum_{n=1}^{n_f} q_n \xi_n + \frac{1}{2} \sum_{n=1}^{n_f} Q_{nn} \xi_n^2 + \sum_{i<n} Q_{in} \xi_i \xi_n,$$

or equivalently

$$y_{\text{m}}(\boldsymbol{\xi}) = q_0 + \mathbf{q}^{\text{T}} \boldsymbol{\xi} + \frac{1}{2} \boldsymbol{\xi}^{\text{T}} \mathbf{Q} \boldsymbol{\xi},$$

with $\mathbf{Q}$ symmetric and $\mathbf{q} = (q_1, \ldots, q_{n_f})^{\text{T}}$. A stationary point $\boldsymbol{\xi}^*$ then satisfies

$$\mathbf{Q}\boldsymbol{\xi}^* = -\mathbf{q},$$

and estimating $\mathbf{q}$ and $\mathbf{Q}$ from observations $y(\boldsymbol{\xi}^i)$ makes it possible to estimate $\boldsymbol{\xi}^*$. The type of the stationary point obtained depends on the sign of the eigenvalues of $\mathbf{Q}$. If all of them are negative, which we shall assume, $\boldsymbol{\xi}^* = -\mathbf{Q}^{-1}\mathbf{q}$ is the (global) maximizer of $y_{\text{m}}(\boldsymbol{\xi})$. Note that this choice amounts to performing one step of the Newton method (Section 4.3.3.3). The operating conditions are often normalized so as to put the centre of $\boldsymbol{\xi}$ at $\mathbf{0}$ and each factor ξ_n between -1 and 1. The experiments used for a second-degree model should have at least three levels for each factor. A central-composite design is often used (Box and Wilson, 1951; Montgomery, 1976). It is obtained from a two-level factorial design by adding central points (in $\mathbf{0}$) and axial points in the directions of the unit vectors $\pm\mathbf{e}^n$ ($n = 1, \ldots, n_f$). These designs have the advantage of being built from those used during the steepest ascent. (See also (Box and Hunter, 1957) for rotatable designs, with constant variance of the prediction of $y_{\text{m}}(\boldsymbol{\xi})$ on spheres centred at $\mathbf{0}$.)

In contrast to the model-free approaches of the previous section, the response-surface methodology provides information on the way the factors influence the response of the system (comparison of the effects of factors, quantification of interactions...). The experimental conditions are, however, specified *a priori*, and one should note that:

— the experiment is designed without taking the objective into account; all parameters receive the same attention, irrespective of their influence on $y_{\text{m}}(\boldsymbol{\xi})$,

— prior knowledge gained from previous studies, if any, is not taken advantage of,

— the approach is intrinsically local; possible constraints on $\boldsymbol{\xi}$ (other than bounds on its components) are not taken into account,

— the model structure is assumed to be linear or quadratic in $\boldsymbol{\xi}$, which may not be true over the whole region of interest,

— non-Bayesian estimation of the parameters requires a large number of measurements (at least equal to the number of parameters).

The definition of a cost function, such as

$$j(\hat{\mathbf{p}}|\mathbf{p}) = y_{\mathrm{m}}[\xi^*(\mathbf{p}), \mathbf{p}] - y_{\mathrm{m}}[\xi^*(\hat{\mathbf{p}}), \mathbf{p}],$$

with

$$\xi^*(\mathbf{p}) = \arg \max_{\xi \in \xi} y_{\mathrm{m}}(\xi, \mathbf{p}),$$

makes it possible to

— take the objective (maximization of $y_{\mathrm{m}}(\xi)$) into account when estimating the parameter vector $\mathbf{p}$ and designing the experiments ξ^i to collect the data for this estimation,

— consider model structures $y_{\mathrm{m}}(\xi, \mathbf{p})$ that may not be polynomial in ξ,

— incorporate prior knowledge through a Bayesian formulation of the problem. Such a Bayesian formulation seems to have been used first by Lindley (1968) and Brooks (1977), in the case where the objective is to reach a given target level $y_{\mathrm{m}}(\xi) = c$.

If $\mathbf{p}$ is estimated by maximum likelihood, a possible criterion for experiment design is L-optimality; see Section 6.1, (Pronzato and Walter, 1992a), as well as (Chatterjee and Mandal, 1981, 1985; Mandal, 1989) for cost functions quadratic in ξ. If the estimator of $\mathbf{p}$ is the minimum-risk estimator (Section 3.5.2), few measurements are necessary, and the choice of the experimental conditions can be performed according to an $\mathrm{L_B}$-optimality criterion; see Section 6.5 and (Pronzato and Walter, 1991c, 1992a, 1992b).

REMARK 4.27

Since the structure of the models employed is usually LP (see, *e.g.*, the previous quadratic models in ξ), their parameters can be estimated by recursive least squares (Section 4.1.4). One may then, after each estimation, design the experiment associated with the next observation, which defines the corresponding regressor vector. This is a sequential-design problem, for which one may refer to the real-time control problem of Section 6.3.2.2. A naive approach (corresponding to *forced certainty equivalence control*) would be to choose

$$\xi^{k+1} = \arg \max_{\xi \in \xi} y_{\mathrm{m}}[\xi, \hat{\mathbf{p}}(k)],$$

with $\hat{\mathbf{p}}(k)$ the estimate of the parameters from the first k observations. This procedure usually *does not converge*, and a detailed analysis of its behaviour in the scalar case where $y_{\mathrm{m}}(\xi, \mathbf{p}) = p_1 + p_2\xi + p_3\xi^2$, as well as a modification that ensures convergence can be found in (Bozin and Zarrop, 1991). A dual-control approach is suggested in (Kulcsár, 1995). ◊

4.5 Conclusions

There are hundreds of optimization methods, and it was out of the question to present all of them. It is not fortuitous that the relaxed Gauss-Newton, Levenberg-Marquardt, quasi-Newton and conjugate-gradient methods are among the best known for unconstrained optimization. Many users have applied them, and most efficient algorithms rely on the same basic principles.

The results largely depend on how the algorithms have been coded, and on the care with which numerical problems have been handled. Big software libraries (IMSL, HARWELL, NAG...) include sophisticated implementation of most methods presented. One should therefore resist the temptation to reinvent the wheel, and use existing subroutines. This will leave more time to think about fundamental questions (which structure to choose for the model, what criterion to optimize, how to collect the data...). It will also make it possible to try a variety of algorithms to find the one that performs best. Only when the results are unsatisfactory is one justified in developing a specific code or modifying an existing one. The information in this chapter should help the reader select algorithms worth trying, and find out possible reasons (and remedies) for any failure.

Note, finally, that the core of most optimization-based parameter-estimation algorithms is a simulator computing the model output (or the prediction error), and possibly sensitivity functions or the evolution of adjoint variables. Very often, the time spent in these simulations makes up most of the computer time required by the optimization. The simulation algorithm should therefore be carefully selected, possibly by comparing the performance of various simulators, and the precision of the computation should be no higher than necessary. For simulation of a fourth-order set of ordinary differential equations, we have, for example, found that the simulator described in (Valko and Vajda, 1984), which also computes the first-order sensitivity functions of the output with respect to the parameters, was about 200 times quicker than a commercial simulator for similar precision, which resulted in the same gain in overall optimization time.

5 Uncertainty

It is not enough, in general, merely to find the best value of the parameters with respect to the criterion chosen. It is also important to evaluate the uncertainty attached to this result, taking into account the uncertainty in the data and the numerical errors. Several methods can be used (and possibly combined) for this purpose. None is without drawbacks, and the problem is unlikely ever to allow any totally satisfactory solution. Here again, the method used should be clearly specified.

5.1 Cost contours in parameter space

Remember that the cost contour at level j_1 is the set of points satisfying

$$j(\mathbf{p}) = j_1,$$

where we assume that the cost j is to be minimized. Techniques for characterizing cost contours can also be used with functions of the parameters other than the cost itself. Indeed, the characterization of a level set (or confidence region) by its boundary $f(\mathbf{p}) = f_\alpha$ obviously corresponds to a cost contour for the function f. A cost contour is generally a hypersurface, which may possibly extend to infinity (for example when some parameters are not identifiable). As Section 5.4 will show, a cost contour may also be a hypervolume (*i.e.* a set of positive measure), when the estimator associated with the criterion is not a point estimator. When dim $\mathbf{p} = 2$, the cost contours are similar to contours on a geographical map; see, *e.g.*, Figures 4.46 and 4.47.

In the vicinity of a local minimum, a second-order Taylor series expansion of the cost can be used (provided that the derivatives exist). If the Hessian of the cost is positive-definite, the cost contours may then be approximated by ellipsoids. Section 5.3.1.4 will present a method for characterizing parameter uncertainty based on this property. For the time being, however, we shall not force an ellipsoidal shape on the cost contours.

5.1.1 Normal noise: cost contours, confidence regions

The method used to build confidence regions depends on whether the noise variance is known *a priori*, unknown or estimated from independent measurements.

5.1.1.1 Noise with known variance

Assume that the prediction error satisfies

$$e_{\mathrm{p}}(t_i, \mathbf{p}^*) = \varepsilon(t_i), \quad i = 1, \ldots, n_t,$$

where the $\varepsilon(t_i)$'s are independent random variables normally distributed $\mathcal{N}(0, \sigma^2)$, with σ^2 known. Maximum-likelihood estimation then corresponds to minimizing

$$j(\mathbf{p}) = \sum_{i=1}^{n_t} [e_{\mathrm{p}}(t_i, \mathbf{p})]^2.$$

When $\mathbf{p}$ is the true value $\mathbf{p}^*$ of the parameter vector, the value of the cost becomes

$$j(\mathbf{p}^*) = \sum_{i=1}^{n_t} [\varepsilon(t_i)]^2 \approx n_t\sigma^2.$$

It is thus pointless to try to reduce the cost below $j_1 = n_t\sigma^2$. The larger the noise variance σ^2, the higher the level of the cost contour one should be content with. The effect of the noise is thus to raise the highest acceptable value of the cost, thereby expanding the set of acceptable models.

The error $\mathbf{e}(\mathbf{p})$ lies in an n_t-dimensional space, and $\mathbf{e}^{\mathrm{T}}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*)/\sigma^2$ has a chi-square distribution with n_t degrees of freedom, denoted by $\chi^2(n_t)$ in what follows. Let $\chi^2_\alpha(n_t)$ be the value with probability α of being exceeded by a random variable having the distribution $\chi^2(n_t)$. It is tabulated and can also be computed (Press *et al.*, 1986). The set

$$\mathbb{R}_1^\alpha = \{\mathbf{p} \in \mathbb{R}^{n_p} \mid \mathbf{e}^{\mathrm{T}}(\mathbf{p})\mathbf{e}(\mathbf{p})/\sigma^2 \leq \chi^2_\alpha(n_t)\}$$

defines a $100(1-\alpha)\%$ confidence region for the parameters. The boundary of this region can be characterized through techniques such as those presented in Section 5.1.2. If the hypotheses on the noise are correct, and if the same experiment is repeated à large number of times, this confidence region will contain the true value $\mathbf{p}^*$ of the parameters in $100(1-\alpha)\%$ of cases. This is a reminder that $\mathbf{p}^*$ may happen to lie outside the confidence region. The smaller α is, the less probable this event becomes (but the larger the region). A common choice is $\alpha = 0.05$.

Assume that the prediction error is in fact an output error, that is

$$e_{\mathrm{p}}(t_i, \mathbf{p}) = y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p}), \; i = 1, \ldots, n_t.$$

The vector $\mathbf{y}^{\mathrm{s}}$ containing all available measurements is a point in the space of observations, as is the vector $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ of associated model outputs. When $\mathbf{p}$ varies, $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ describes a hypersurface $\mathbb{S}_{\mathrm{exp}}$ in this space, the *expectation surface* or *solution locus*. For LP model structures, it is a hyperplane (*i.e.* an n_{p}-dimensional plane), whereas for non-LP model structures it is generally a curved hypersurface; see, *e.g.*, Figure 4.45. Two factors may contribute to making a model structure non-LP (Bates and Watts, 1980, 1988; Ratkowsky, 1983; Seber and Wild, 1989). The first is curvature of $\mathbb{S}_{\text{exp}}$, which corresponds to the *intrinsic nonlinearity*. The second expresses how a uniform grid in parameter space maps into a non-uniform grid on $\mathbb{S}_{\text{exp}}$ and corresponds to the *parametric nonlinearity*. Thus, for instance, the nonlinearity in p of the scalar model structure

$$y_{\mathrm{m}}(t, p) = p^2 u(t)$$

is only due to the parametrization, and is not intrinsic. Modification of the parametrization may sometimes reduce nonlinearity of the model structure in its parameters to intrinsic nonlinearity, although the situation is generally far more complicated than in the simple example above (Bates and Watts, 1981; Hamilton, Watts and Bates, 1982).

Let the matrix $\mathbf{\Pi}(\mathbf{p})$ be the orthogonal projector onto the tangent plane to $\mathbb{S}_{\text{exp}}$ at $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$ (Figure 5.1), given by

$$\mathbf{\Pi}(\mathbf{p}) = \frac{\partial \mathbf{y}^{\mathrm{m}}(\mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}} \left\{ \left[\frac{\partial \mathbf{y}^{\mathrm{m}}(\mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}\right]^{\mathrm{T}} \left[\frac{\partial \mathbf{y}^{\mathrm{m}}(\mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}\right] \right\}^{-1} \left[\frac{\partial \mathbf{y}^{\mathrm{m}}(\mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}\right]^{\mathrm{T}}.$$

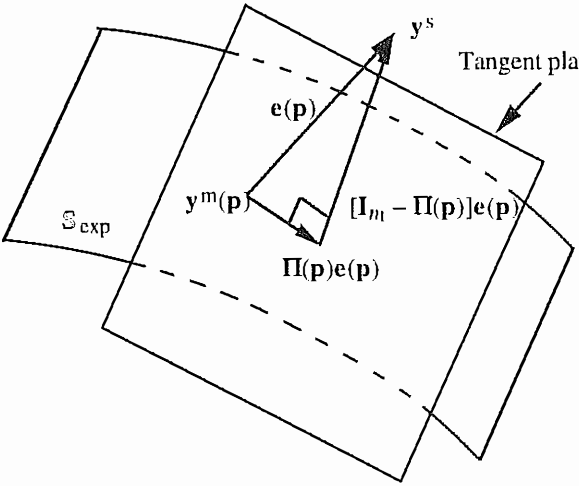

$\mathbf{y}^{\mathrm{s}}$
Tangent plane to $\mathbb{S}_{\text{exp}}$ at $\mathbf{y}^{\mathrm{m}}(\mathbf{p})$
$\mathbf{e}(\mathbf{p})$
$\mathbf{y}^{\mathrm{m}}(\mathbf{p})$
$[\mathbf{I}_{n_{\mathrm{t}}} - \mathbf{\Pi}(\mathbf{p})]\mathbf{e}(\mathbf{p})$
$\mathbb{S}_{\text{exp}}$
$\mathbf{\Pi}(\mathbf{p})\mathbf{e}(\mathbf{p})$

Figure 5.1. Expectation surface $\mathbb{S}_{\text{exp}}$ for a non-LP model structure

If $n_{\mathrm{t}} = \dim \mathbf{y}^{\mathrm{s}}$, $n_{\mathrm{p}} = \dim \mathbf{p}$, $\mathbf{e}(\mathbf{p}) = \mathbf{y}^{\mathrm{s}} - \mathbf{y}^{\mathrm{m}}(\mathbf{p})$ and $\mathbf{p}^*$ is the true value of the parameters, then

— $\mathbf{e}^{\mathrm{T}}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*)/\sigma^2$ has a $\chi^2(n_{\mathrm{t}})$ distribution, as already mentioned;
— $\mathbf{e}^{\mathrm{T}}(\mathbf{p}^*)\mathbf{\Pi}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*)/\sigma^2$ has a $\chi^2(n_{\mathrm{p}})$ distribution, as the projection of the error onto an n_{p}-dimensional hyperplane;
— $\mathbf{e}^{\mathrm{T}}(\mathbf{p}^*)[\mathbf{I}_{n_{\mathrm{t}}} - \mathbf{\Pi}(\mathbf{p}^*)]\mathbf{e}(\mathbf{p}^*)/\sigma^2$ has a $\chi^2(n_{\mathrm{t}} - n_{\mathrm{p}})$ distribution, for it complements the projection above.

If $\mathbb{S}_{\text{exp}}$ is flat (*i.e.* its intrinsic curvature is zero), $\mathbf{e}^{\mathrm{T}}(\mathbf{p}^*)[\mathbf{I}_{n_{\mathrm{t}}} - \mathbf{\Pi}(\mathbf{p}^*)]\mathbf{e}(\mathbf{p}^*)$ is constant (and thus cannot be used to define a confidence region). The confidence regions $\mathbb{R}_{\mathrm{I}}^{\alpha}$ and

$$\mathbb{R}_2^{\alpha} = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid \mathbf{e}^\mathrm{T}(\mathbf{p})\mathbf{\Pi}(\mathbf{p})\mathbf{e}(\mathbf{p})/\sigma^2 \le \chi^2_{\alpha}(n_\mathrm{p})\}$$

are then both defined by least-squares cost contours.

EXAMPLE 5.1

Consider the model

$$y_\mathrm{m}(\boldsymbol{\xi}, \mathbf{p}) = p_1\xi_1 + p_2\xi_2 + p_1^3(1 - \xi_1) + p_2^2(1 - \xi_2),$$

and data generated by computing $y_\mathrm{m}(\boldsymbol{\xi}, \mathbf{p}^*)$ at $\mathbf{p}^* = (1, 2)^\mathrm{T}$ and adding realizations of independent errors distributed $\mathcal{N}(0, 1)$. Assume first that three data points

$$\mathbf{y}^\mathrm{s} = (5.673, 3.362, 3.043)^\mathrm{T}$$

have been collected under the experimental conditions defined by

$$\boldsymbol{\xi}^1 = (1, 0)^\mathrm{T} \quad \text{and} \quad \boldsymbol{\xi}^2 = \boldsymbol{\xi}^3 = (1, 1)^\mathrm{T}.$$

Figure 5.2 presents the confidence regions $\mathbb{R}_1^{0.05}$ (solid line) and $\mathbb{R}_2^{0.05}$ (dashed line) for the model parameters, computed for $\sigma^2 = 1$. We have already noticed in Section 4.3.9.1 (Example 4.21) that $\mathbf{p}$ is only locally identifiable under these experimental conditions. We have also seen that they make $\mathbb{S}_\mathrm{exp}$ flat. The true value $\mathbf{p}^*$ of the parameter vector is indicated by a cross.

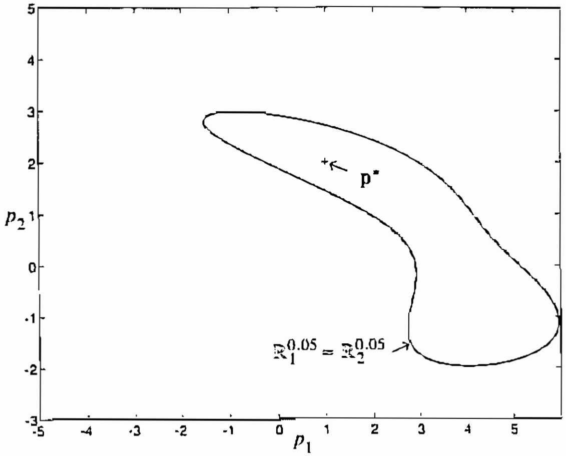


Figure 5.2. Confidence regions $\mathbb{R}_1^{0.05}$ (solid line) and $\mathbb{R}_2^{0.05}$ (dashed line) for Example 5.1 (σ^2 known), when $\mathbb{S}_\mathrm{exp}$ is flat

Assume next that the experimental conditions are

$$\xi_1 = (1, 0)^{\mathrm{T}}, \quad \xi_2 = (1, 1)^{\mathrm{T}} \quad \text{and} \quad \xi_3 = (0, 1)^{\mathrm{T}},$$

so the parameter vector $\mathbf{p}$ becomes globally identifiable. With the same noise realizations as previously, the observations are identical to the previous case:

$$\mathbf{y}^{\mathrm{s}} = (5.673, 3.362, 3.043)^{\mathrm{T}}.$$

Figure 5.3 presents the associated confidence regions $\mathbb{R}_1^{0.05}$ (solid line) and $\mathbb{R}_2^{0.05}$ (dashed line), computed for $\sigma^2 = 1$, and illustrates the fact that these regions may be disconnected when $\mathbb{S}_{\mathrm{exp}}$ is curved (see Figures 4.45 and 4.46). The true value $\mathbf{p}^*$ of the parameter vector is indicated by a cross. ◊

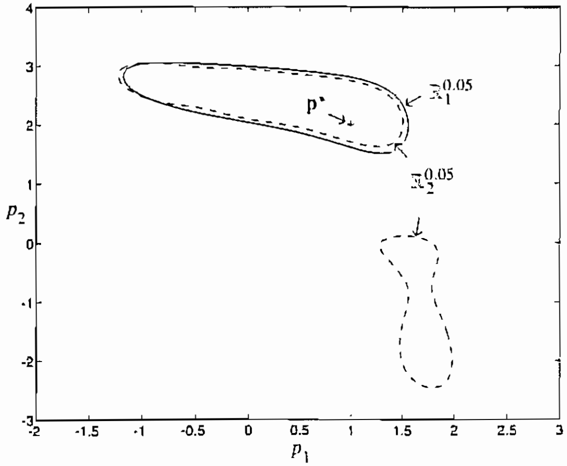


Figure 5.3. Confidence regions $\mathbb{R}_1^{0.05}$ (solid line) and $\mathbb{R}_2^{0.05}$ (dashed line) for Example 5.1 (σ^2 known) when $\mathbb{S}_{\mathrm{exp}}$ is not flat

5.1.1.2 Noise with unknown variance

In this section, we assume that σ^2 is unknown and cannot be estimated independently from replicated measurements. (The case where σ^2 is estimated independently will be considered in Section 5.1.1.3.) $\mathbb{R}_1^{0.05}$ and $\mathbb{R}_2^{0.05}$ can therefore no longer be used. However, $\mathbf{e}^{\mathrm{T}}(\mathbf{p}^*)[\mathbf{I}_{n_t} - \mathbf{\Pi}(\mathbf{p}^*)]\mathbf{e}(\mathbf{p}^*)$ and $\mathbf{e}^{\mathrm{T}}(\mathbf{p}^*)\mathbf{\Pi}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*)$ are independent, so

$$\frac{\mathbf{e}^{\mathrm{T}}(\mathbf{p}^*)\mathbf{\Pi}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*)}{\mathbf{e}^{\mathrm{T}}(\mathbf{p}^*)[\mathbf{I}_{n_t} - \mathbf{\Pi}(\mathbf{p}^*)]\mathbf{e}(\mathbf{p}^*)} \, \frac{n_t - n_p}{n_p},$$

which does not depend on σ^2, has a Fisher-Snedecor distribution with n_p and $(n_t - n_p)$ degrees of freedom, denoted in what follows by $\mathcal{F}(n_p, n_t - n_p)$. The value having the probability α of being exceeded by a random variable distributed $\mathcal{F}(n_p, n_t - n_p)$ will be denoted by $F_\alpha(n_p, n_t - n_p)$. It is tabulated and also easily computed (Press *et al.*, 1986).

The set

$$\mathbb{R}_3^\alpha = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid \frac{\mathbf{e}^\mathrm{T}(\mathbf{p})\boldsymbol{\Pi}(\mathbf{p})\mathbf{e}(\mathbf{p})}{\mathbf{e}^\mathrm{T}(\mathbf{p})[\mathbf{I}_{n_\mathrm{t}} - \boldsymbol{\Pi}(\mathbf{p})]\mathbf{e}(\mathbf{p})} \frac{n_\mathrm{t} - n_\mathrm{p}}{n_\mathrm{p}} \leq F_\alpha(n_\mathrm{p}, n_\mathrm{t} - n_\mathrm{p})\}$$

thus defines a $100(1 - \alpha)\%$ confidence region for the parameters. We can also write

$$\frac{\mathbf{e}^\mathrm{T}(\mathbf{p}^*)\boldsymbol{\Pi}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*)}{\mathbf{e}^\mathrm{T}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*)} = \frac{\mathbf{e}^\mathrm{T}(\mathbf{p}^*)\boldsymbol{\Pi}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*)}{\mathbf{e}^\mathrm{T}(\mathbf{p}^*)\boldsymbol{\Pi}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*) + \mathbf{e}^\mathrm{T}(\mathbf{p}^*)[\mathbf{I}_{n_\mathrm{t}} - \boldsymbol{\Pi}(\mathbf{p}^*)]\mathbf{e}(\mathbf{p}^*)}$$

$$= \frac{1}{1 + \dfrac{\mathbf{e}^\mathrm{T}(\mathbf{p}^*)[\mathbf{I}_{n_\mathrm{t}} - \boldsymbol{\Pi}(\mathbf{p}^*)]\mathbf{e}(\mathbf{p}^*)}{\mathbf{e}^\mathrm{T}(\mathbf{p}^*)\boldsymbol{\Pi}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*)}},$$

so the set of parameter vectors $\mathbf{p}$ such that

$$f(\mathbf{p}) = \frac{\mathbf{e}^\mathrm{T}(\mathbf{p})\boldsymbol{\Pi}(\mathbf{p})\mathbf{e}(\mathbf{p})}{\mathbf{e}^\mathrm{T}(\mathbf{p})\mathbf{e}(\mathbf{p})} \leq \frac{\dfrac{n_\mathrm{p}}{n_\mathrm{t} - n_\mathrm{p}} F_\alpha(n_\mathrm{p}, n_\mathrm{t} - n_\mathrm{p})}{1 + \dfrac{n_\mathrm{p}}{n_\mathrm{t} - n_\mathrm{p}} F_\alpha(n_\mathrm{p}, n_\mathrm{t} - n_\mathrm{p})}$$

coincides with $\mathbb{R}_3^\alpha$ (Halperin, 1963; Hamilton, Watts and Bates, 1982). $\mathbb{R}_3^\alpha$ could equivalently be defined as

$$\mathbb{R}_3^\alpha = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid \frac{\mathbf{e}^\mathrm{T}(\mathbf{p})[\mathbf{I}_{n_\mathrm{t}} - \boldsymbol{\Pi}(\mathbf{p})]\mathbf{e}(\mathbf{p})}{\mathbf{e}^\mathrm{T}(\mathbf{p})\boldsymbol{\Pi}(\mathbf{p})\mathbf{e}(\mathbf{p})} \frac{n_\mathrm{p}}{n_\mathrm{t} - n_\mathrm{p}} \geq F_\alpha(n_\mathrm{t} - n_\mathrm{p}, n_\mathrm{p})\}$$

or

$$\mathbb{R}_3^\alpha = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid \frac{\mathbf{e}^\mathrm{T}(\mathbf{p})\mathbf{e}(\mathbf{p})}{\mathbf{e}^\mathrm{T}(\mathbf{p})\boldsymbol{\Pi}(\mathbf{p})\mathbf{e}(\mathbf{p})} \geq 1 + \frac{n_\mathrm{t} - n_\mathrm{p}}{n_\mathrm{p}} F_\alpha(n_\mathrm{t} - n_\mathrm{p}, n_\mathrm{p})\}.$$

REMARKS 5.1

— The left-hand side of any of the inequalities used to define $\mathbb{R}_3^\alpha$ does not correspond to the cost j minimized in a maximum-likelihood approach. The boundary of the associated confidence region is therefore not a cost contour for j.

— Compared to techniques based on asymptotic properties of the likelihood ratio (Eadie *et al.*, 1971), this approach has the advantage of yielding a nonasymptotic region, valid even when n_t is small.

— $100(1 - \alpha)\%$ confidence regions are not unique. For instance,

$$\mathbb{R}_4^\alpha = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid \frac{n_\mathrm{p}}{n_\mathrm{t} - n_\mathrm{p}} \frac{\mathbf{e}^\mathrm{T}(\mathbf{p})[\mathbf{I}_{n_\mathrm{t}} - \boldsymbol{\Pi}(\mathbf{p})]\mathbf{e}(\mathbf{p})}{\mathbf{e}^\mathrm{T}(\mathbf{p})\boldsymbol{\Pi}(\mathbf{p})\mathbf{e}(\mathbf{p})} \leq F_{(1-\alpha)}(n_\mathrm{t} - n_\mathrm{p}, n_\mathrm{p})\}$$

is also a $100(1 - \alpha)\%$ confidence region for the parameters. (However, this region is unbounded and does not contain the least-squares estimate of the parameters.) One may also choose *a priori* the shape of the region, *e.g.* an ellipsoid centred at $\hat{\mathbf{p}}$, or a strip of parameter space bounded by two parallel hyperplanes. See, *e.g.*, (Dasgupta, 1991) for a discussion on minimum-volume confidence regions. ◊

EXAMPLE 5.1 (continued)

Consider again the model

$$y_{\mathrm{m}}(\boldsymbol{\xi}, \mathbf{p}) = p_1\xi_1 + p_2\xi_2 + p_1^3(1-\xi_1) + p_2^2(1-\xi_2),$$

with

$$\boldsymbol{\xi}^1 = (1, 0)^{\mathrm{T}},\ \boldsymbol{\xi}^2 = \boldsymbol{\xi}^3 = (1, 1)^{\mathrm{T}},\ \mathbf{y}^{\mathrm{s}} = (5.673, 3.362, 3.043)^{\mathrm{T}}.$$

Figure 5.4 shows $\mathbb{R}_3^{0.05}$, together with $\mathbb{R}_1^{0.05}$ as in Figure 5.2.

At the scale of this picture, $\mathbb{R}_1^{0.05}$ and $\mathbb{R}_2^{0.05}$ are indistinguishable. $\mathbb{R}_3^{0.05}$ is larger than $\mathbb{R}_1^{0.05}$, a price to be paid for not knowing σ^2.

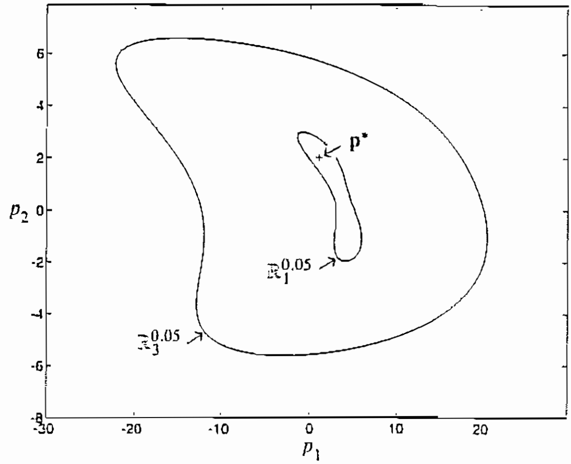


Figure 5.4. Confidence regions $\mathbb{R}_1^{0.05}$ and $\mathbb{R}_3^{0.05}$ for Example 5.1 with $\mathbb{S}_{\mathrm{exp}}$ flat

Curvature of $\mathbb{S}_{\mathrm{exp}}$ may produce regions with complicated shapes. Figure 5.5 presents $\mathbb{R}_3^{0.05}$ when $\mathbf{y}^{\mathrm{s}} = (5.673, 3.362, 3.043)^{\mathrm{T}}$, $\boldsymbol{\xi}^1 = (1, 0)^{\mathrm{T}}$, $\boldsymbol{\xi}^2 = (1, 1)^{\mathrm{T}}$ and $\boldsymbol{\xi}^3 = (0, 1)^{\mathrm{T}}$. ◊

5.1.1.3 Noise with independently estimated variance

If the value of σ^2 is unknown but can be estimated *independently* by n_{e} repetitions of the same experiment at t_i (Example 3.1), its estimate $\hat{\sigma}^2$ given by

$$\hat{\sigma}^2 = \frac{1}{n_{\mathrm{e}} - 1} \sum_{k=1}^{n_{\mathrm{e}}} [y(t_{ik}) - m_i]^2, \qquad \text{with} \quad m_i = \frac{1}{n_{\mathrm{e}}} \sum_{k=1}^{n_{\mathrm{e}}} y(t_{ik}),$$

is such that $(n_{\mathrm{e}} - 1)\hat{\sigma}^2/\sigma^2$ has a χ^2 distribution with $(n_{\mathrm{e}} - 1)$ degrees of freedom.

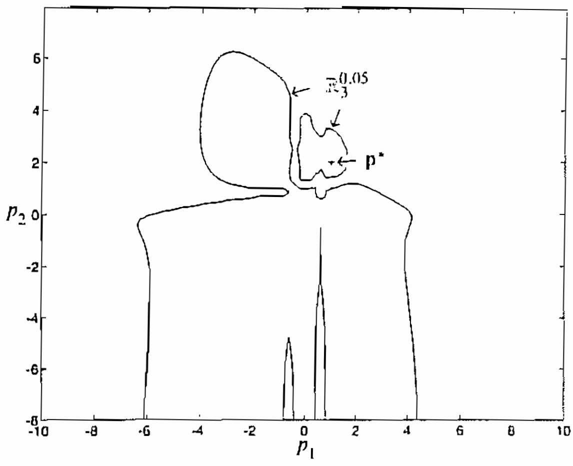


Figure 5.5. Confidence regions $\mathbb{R}_3^{0.05}$ for Example 5.1 when $\mathbb{S}_{\mathrm{exp}}$ is not flat

The random variable

$$\frac{\mathbf{e}^{\mathrm{T}}(\mathbf{p}^*)\mathbf{\Pi}(\mathbf{p}^*)\mathbf{e}(\mathbf{p}^*)}{n_{\mathrm{p}}\hat{\sigma}^2}$$

then has an $\mathcal{F}(n_{\mathrm{p}}, n_{\mathrm{c}} - 1)$ distribution, which can be used to define a confidence region (Hamilton, Watts and Bates, 1982)

$$\mathbb{R}_5^{\alpha} = \{\mathbf{p} \in \mathbb{R}^{n_{\mathrm{p}}} \mid \frac{\mathbf{e}^{\mathrm{T}}(\mathbf{p})\mathbf{\Pi}(\mathbf{p})\mathbf{e}(\mathbf{p})}{n_{\mathrm{p}}\hat{\sigma}^2} \leq F_{\alpha}(n_{\mathrm{p}}, n_{\mathrm{c}} - 1)\}.$$

5.1.2 Determination of points on a cost contour

Many methods can be employed to draw cost contours in a two-dimensional space. One may, for instance, compute the cost at each node of a grid covering the region of interest, then use some standard graphic routine to transform the result into a series of approximate cost contours. This is the procedure followed for Figures 5.2 to 5.5. One may also try to avoid gridding and attempt to follow a cost contour with a step size that is varied according to the boundary curvature (Norton and Veres, 1991). These two approaches easily extend to three-dimensional problems by considering series of two-dimensional cross sections that can be drawn in perspective. A third possible approach is by random scanning. Assume that a vector $\hat{\mathbf{p}}$ with $j(\hat{\mathbf{p}}) < j_1$ has been obtained as a result of the optimization stage. Leaving $\hat{\mathbf{p}}$ along $\mathbf{d}$, we find a vector $\mathbf{p}_1$ such that $j(\mathbf{p}_1) = j_1$. A set of points on the contour for cost j_1 will then be obtained by varying the direction $\mathbf{d}$.

First stage. Find $\mathbf{p}_2$ along $\mathbf{d}$, *i.e.* $\mathbf{p}_2 = \hat{\mathbf{p}} + \lambda\mathbf{d}$, such that $j(\mathbf{p}_2) > j_1$, by increasing the scalar step length λ until $j(\mathbf{p}_2) > j_1$. Provided the cost is continuous, there exists at least one point $\mathbf{p}_1$ on the line segment between $\hat{\mathbf{p}}$ and $\mathbf{p}_2$ such that $j(\mathbf{p}_1) = j_1$.

Second stage. Define $\mathbf{p}(\mu) = \mu\hat{\mathbf{p}} + (1 - \mu)\mathbf{p}_2$. Since $\mathbf{p}_1$ is on the line segment joining $\hat{\mathbf{p}}$ and $\mathbf{p}_2$, finding $\mathbf{p}_1$ amounts to finding $\hat{\mu}$ with $0 < \hat{\mu} < 1$ such that the function

$$f(\mu) = j[\mathbf{p}(\mu)] - j_1$$

is zero (Figure 5.6). A particularly efficient method is dichotomy (Section 4.3.2.2). At each iteration, f is evaluated at the middle of the remaining feasible interval for $\hat{\mu}$. If $f > 0$ the left half of the interval is deleted, if $f = 0$ a point on the cost contour has been found, and if $f < 0$ the right half is deleted. The interval containing $\hat{\mu}$ is thus halved at each iteration. Its length therefore decreases very rapidly, and it is recommended that the number of iterations required (for a given precision and a given initial interval) be calculated in advance, to avoid unnecessary iterations. Remember that evaluation of f at a given μ requires simulation of the associated model so as to compute $\mathbf{y}^{\mathrm{m}}[\mathbf{p}(\mu)]$ and then $j[\mathbf{p}(\mu)]$.

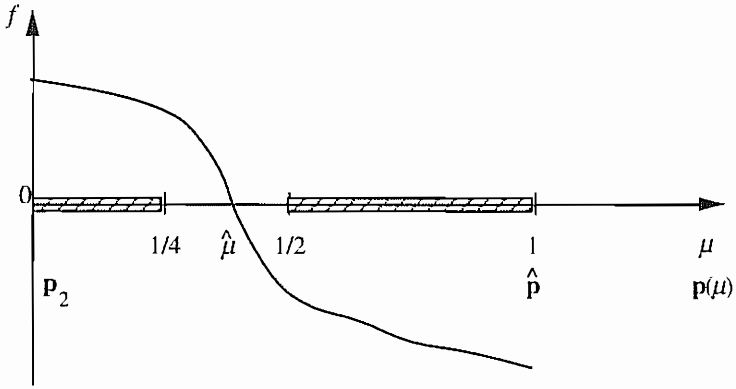


Figure 5.6. Determination of a point on a cost contour by dichotomy

If the cost contour cuts the line segment considered several times, dichotomy will only locate one of the intersections.

By varying the direction $\mathbf{d}$, one gets a cloud of points $\{\mathbf{p}_{\mathrm{cc}}\}$ on the cost contour at level j_1. This is like a speleologist exploring a cave with a torch. If the exploration is always from $\hat{\mathbf{p}}$, *i.e.* if the speleologist does not move, some parts of a non-convex cost contour (the cave wall) may remain in the shade, giving an over-optimistic view of parameter uncertainty. The origin of the exploration must therefore be moved (Richalet, Rault and Pouliquen, 1971). A possible policy is to use as successive origins for exploration the points in the cloud with the largest or smallest ith components of $\mathbf{p}$ ($i = 1, \ldots, \dim \mathbf{p}$). This policy favours the extreme points of the cost contour in each axis direction, which is important when accurate uncertainty intervals for the parameters are sought.

5.1.3 Characterization of non-connected domains

The set of vectors $\mathbf{p}$ such that $j(\mathbf{p}) < j_1$ may not be connected. This may result from lack of identifiability of the structure, in which case the problem can often be solved by characterizing the set of all parameter vectors giving the model the same input-output behaviour. With any point of the cloud obtained as above, one can then associate others on the same cost contour (Figure 5.7). See Example 4.21 and Section 5.4.2.2.

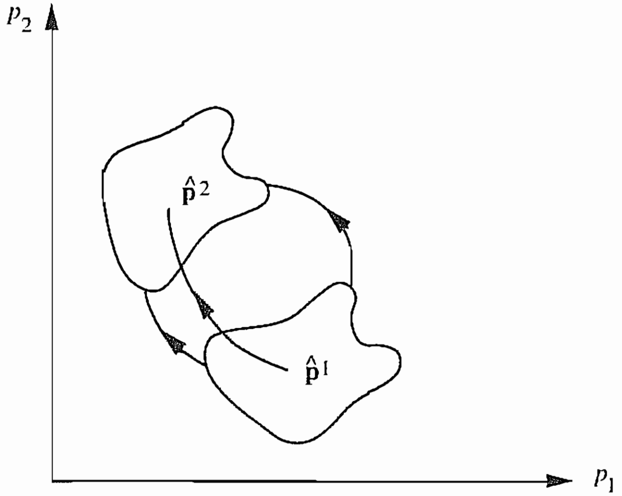


Figure 5.7. Characterization of non-connected cost contours *via* identifiability studies; each parameter value $\hat{\mathbf{p}}^2$ gives the same input-output behaviour as the corresponding $\hat{\mathbf{p}}^1$

Sometimes, however, the non-connectedness of the uncertainty set cannot be detected by an identifiability analysis. If the cost contour corresponds to the cost function j, the problem relates to inverse multimodality of j and may be avoided by a suitable choice of experimental conditions (Section 4.3.9.1). Another approach consists in characterizing the uncertainty domain obtained so far by a simple outer set (*e.g.*, a union $\mathbb{U}$ of orthotopes), and performing a new optimization of j in the complement of $\mathbb{U}$ in $\mathbb{P}$. The global optimization algorithm of Section 4.3.9.2 may be used for that purpose. If a new value $\hat{\mathbf{p}}$ is found, such that $j(\hat{\mathbf{p}}) < j_1$, a local characterization of the cost contour at level j_1 is performed starting at $\hat{\mathbf{p}}$, and used to update $\mathbb{U}$. If no such $\hat{\mathbf{p}}$ can be found, the search is terminated. Interval analysis can also be employed to characterize cost contours approximately but in a global and guaranteed way (Didrit, Jaulin and Walter, 1995).

5.1.4 Representation of cost contours

As long as the number of parameters is less than four, direct depiction of a cloud of points $\{\mathbf{p}_{cc}\}$ on the cost contour can be used. (The same kind of technique can also be used in the context of bounded-error parameter estimation considered in Section 5.4

(Norton, 1986b).) When the number of parameters is larger than three, several policies can be used.

— The cloud of points may be projected onto sub-spaces (Richalet, 1991). One may, for instance, project it onto the axes of parameter space, and thus get *parameter uncertainty intervals* (PUI's)

$$p_{i_{\min}} = \min_{\{\mathbf{p}_{\mathrm{cc}}\}} p_i \quad \text{and} \quad p_{i_{\max}} = \max_{\{\mathbf{p}_{\mathrm{cc}}\}} p_i, \quad i = 1, \ldots, n_{\mathrm{p}}.$$

Note that when the cloud of points $\{\mathbf{p}_{\mathrm{cc}}\}$ is for a 95% confidence region, the parameter uncertainty intervals thus obtained are not the smallest 95% confidence intervals obtainable (Figure 5.8).

— The cloud of points may be approximated by a quadratic surface, the equation of which has to be determined (Richalet, Rault and Pouliquen, 1971). See also the robust estimation of correlation coefficients through ellipsoidal trimming (Titterington, 1978).

— *Principal component analysis* may be used to study the properties of the cloud (Jackson, 1991), and indicate possible correlations between parameters.

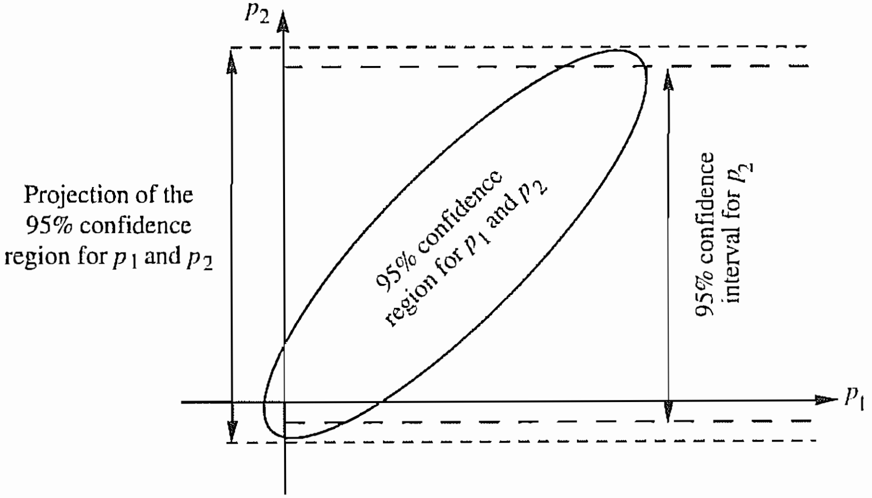

p_2
Projection of the 95% confidence region for p_1 and p_2
95% confidence region for p_1 and p_2
95% confidence interval for p_2
p_1

Figure 5.8. The projection of a 95% confidence region for p_1 and p_2 onto the p_2 axis is not the smallest 95% confidence interval for p_2

Whatever approach is taken, the determination of cost contours will require a large number of model runs (*simulations*) if a realistic view of parameter uncertainty is to be provided (and the larger dim $\mathbf{p}$, the larger this number).

5.2 Monte-Carlo methods

5.2.1 Principle

Parameter estimation using data collected from a system can be summarized as follows:

$$\text{System} \xrightarrow{\text{Experiment}} \text{Data } \mathbf{y}^{\mathrm{s}} \xrightarrow{\text{Estimation}} \text{Estimate } \hat{\mathbf{p}}.$$

Repetition of identical experiments will generally not yield the same results, because of the perturbations acting on the system and noise corrupting the measurements. Before the data are collected, $\mathbf{y}^{\mathrm{s}}$ is thus a random vector, and so is the associated estimator $\hat{\mathbf{p}}(\mathbf{y}^{\mathrm{s}})$:

$$\text{Random vector } \mathbf{y}^{\mathrm{s}} \xrightarrow{\text{Estimation}} \text{Estimator } \hat{\mathbf{p}}(\mathbf{y}^{\mathrm{s}}).$$

The measurements make up a particular realization of the random vector $\mathbf{y}^{\mathrm{s}}$, with which a particular estimate $\hat{\mathbf{p}}(\mathbf{y}^{\mathrm{s}})$ is associated. Monte-Carlo methods aim to determine statistical characteristics of the population of estimates yielded by the set of all possible realizations of $\mathbf{y}^{\mathrm{s}}$ (*i.e.* all possible experimental results). *Fictitious data vectors* $\mathbf{y}^{\mathrm{sf}}$ are generated for this purpose, by running the model for the estimated value of the parameters, incorporating realizations of random variables to represent the influence of perturbations and noise:

$$\text{Run of model } M(\hat{\mathbf{p}}) \xrightarrow{\text{Perturbations}} \text{Fictitious data } \mathbf{y}^{\mathrm{sf}}.$$

Each vector of fictitious data gives a fictitious estimate $\hat{\mathbf{p}}^{\mathrm{f}} = \hat{\mathbf{p}}(\mathbf{y}^{\mathrm{sf}})$, calculated as for real data. A set of fictitious estimates can thus be constructed,

$$\{\mathbf{y}^{\mathrm{sf}}\} \xrightarrow{\text{Estimation}} \{\hat{\mathbf{p}}^{\mathrm{f}}\},$$

the statistical properties of which can be studied. Most often, $\hat{\mathbf{p}}^{\mathrm{f}}$ is taken to be a normal random vector, and its distribution is simply characterized by the empirical mean and covariance matrix of the fictitious estimates. This method thus requires a large number of *estimations* (and *a fortiori* of model runs). Various techniques have been suggested to reduce the volume of computation required; see, *e.g.*, (Grant and Solberg, 1983).

The generation of credible fictitious data requires a realistic model for the perturbations. Assume for instance the overall model under study to be an ARARMAX, as shown in Figure 5.9, where the ε's are i.i.d. $\mathcal{N}(0, \sigma^2)$, with σ^2 unknown. The Monte-Carlo method then leads to the following procedure, which can easily be adapted to other model structures incorporating a description of the perturbations acting on the system or noise corrupting the measurements.

— From the actual data, compute an estimate $\hat{\mathbf{p}}$ of the vector of unknown coefficients in the polynomials A, B, C and D and an estimate $\hat{\sigma}^2$ for the variance of $\varepsilon(t)$ (*e.g.*, using the conditional maximum-likelihood method, based on the prediction error $e_p(t, \mathbf{p})$; see Section 3.3.2).

— Check that the resulting prediction errors $e_p(t, \hat{\mathbf{p}})$ do not blatantly contradict the hypotheses on the noise (Chapter 7).

— Generate vectors of fictitious data by running the model with parameters $\hat{\mathbf{p}}$ for various realizations of i.i.d. $\mathcal{N}(0, \hat{\sigma}^2)$ variables $\hat{\varepsilon}(t)$.

— Estimate the parameters of the polynomials A, B, C and D from each of these vectors of fictitious data, using the same method as for the actual data.

— Estimate the mean and covariance matrix of the fictitious parameter estimates thus obtained (or, more simply, estimate the standard deviation for each component of $\mathbf{p}$).

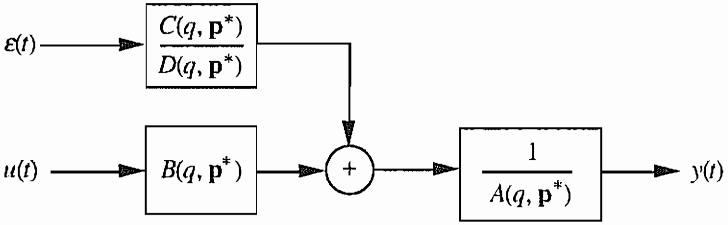

$\varepsilon(t)$
$\frac{C(q, \mathbf{p}^*)}{D(q, \mathbf{p}^*)}$
$u(t)$
$B(q, \mathbf{p}^*)$
+
$\frac{1}{A(q, \mathbf{p}^*)}$
$y(t)$

Figure 5.9. ARARMAX, with the ε's i.i.d. $\mathcal{N}(0, \sigma^2)$

5.2.2 Number of significant digits of the estimate: the CESTAC method

The CESTAC method, already mentioned in connection with stopping rules for iterative algorithms (Section 4.3.7), can also be used to evaluate the combined effect of uncertainty in the data and numerical errors on the number of significant digits of the estimate. The procedure can be summarized as:

— Obtain vectors $\mathbf{y}^{\text{sf}}$ of fictitious data by a classical Monte-Carlo method.

— For each of them, apply the CESTAC method to perturb the computation of the corresponding fictitious estimate of the parameter vector, by random addition of least significant bits to intermediate results. (Note that the effect of perturbation by noise on the data is usually much stronger.)

— For each parameter, use the empirical mean and variance of the fictitious estimates to evaluate its number of significant digits. ,

This number of significant digits is a rather coarse characterization of precision, which can be obtained with only a small number of fictitious data sets (typically three).

5.2.3 Generating fictitious data by jack-knife and bootstrap

One difficulty with Monte-Carlo methods lies in the choice of the distribution used to generate the fictitious data $\mathbf{y}^{\text{sf}}$. The *jack-knife* (Quenouille, 1949) and *bootstrap* (Efron,

1982) methods make it possible to avoid estimating the distribution of the noise from the residuals.

5.2.3.1 Jack-knife

Let $\hat{\mathbf{p}}$ be the estimate obtained from all the data $\mathbf{y}^{\mathrm{s}}$ and let $\hat{\mathbf{p}}_{-i}$ $(i = 1, \dots, n_{\mathrm{t}})$ be the estimate obtained from all the data but the ith. To compensate for the vectors used to estimate $\hat{\mathbf{p}}$ and $\hat{\mathbf{p}}_{-i}$ having $(n_{\mathrm{t}} - 1)$ common elements, which makes $\hat{\mathbf{p}}$ and $\hat{\mathbf{p}}_{-i}$ artificially close, one defines n_{t} pseudo-estimates by

$$\hat{\mathbf{p}}_{(i)} = \hat{\mathbf{p}} + (n_{\mathrm{t}} - 1)(\hat{\mathbf{p}}_{-i} - \hat{\mathbf{p}}), \quad i = 1, \dots, n_{\mathrm{t}},$$

and computes the mean and covariance matrix of the population of the $\hat{\mathbf{p}}_{(i)}$'s, from which confidence intervals can be obtained (Seber and Wild, 1989; Wonnacott and Wonnacott, 1984). The main advantage of this approach lies in its simplicity. It seems, however, less flexible and reliable than the bootstrap method (Diaconis and Efron, 1983).

5.2.3.2 Bootstrap

The bootstrap method (see (DiCiccio and Romano, 1988; Hinkley, 1988) for more details) uses only the data $\mathbf{y}^{\mathrm{s}}$ and model $M(\hat{\mathbf{p}})$. The errors are assumed to be independent random variables, with identical but otherwise unspecified distribution. Assume for instance that

$$y(t_i) = y_{\mathrm{m}}(t_i, \mathbf{p}^*) + b_i, \quad i = 1, \dots, n_{\mathrm{t}},$$

where the b_i's correspond to i.i.d. random variables. An estimate of b_i is provided by the ith residual:

$$\hat{b}_i = y(t_i) - y_{\mathrm{m}}(t_i, \hat{\mathbf{p}}), \quad i = 1, \dots, n_{\mathrm{t}},$$

where $\hat{\mathbf{p}}$ is the estimate of $\mathbf{p}^*$. A vector $\mathbf{y}^{\mathrm{sf}}$ of fictitious data $y^{\mathrm{f}}(t_i)$ is then obtained as

$$y^{\mathrm{f}}(t_i) = y_{\mathrm{m}}(t_i, \hat{\mathbf{p}}) + \hat{b}, \quad i = 1, \dots, n_{\mathrm{t}},$$

where, for each t_i, $\hat{b}$ is randomly chosen among the residuals $\hat{b}_k$ $(k = 1, \dots, n_{\mathrm{t}})$ considered as equiprobable. This amounts to substituting the empirical distribution of the residuals for the true distribution of the b_i's, which seems the more acceptable the closer $\hat{\mathbf{p}}$ is to $\mathbf{p}^*$. Repeating this operation, one obtains a population of vectors of fictitious data, from which a population of parameter estimates can be derived. The characteristics of this population (mean, covariance matrix...) can then be studied.

When $\hat{\mathbf{p}}$ is obtained by minimizing a cost based on a prediction error, the residuals $\hat{b}_i$ are simply replaced by the prediction errors computed at $\hat{\mathbf{p}}$.

5.3 Methods based on the density of the estimator

First, we shall use a bound on the covariance matrix of the parameter estimates to characterize their uncertainty. Two cases will be considered. In Section 5.3.1, no prior information on the parameters is available, and they are regarded as unknown but deterministic quantities. This corresponds to the use of non-Bayesian estimators, such as those presented in Sections 3.1 to 3.3. In Section 5.3.2, the prior distribution of the parameters, regarded as random variables, is assumed to be known. This situation allows use of the Bayesian estimators presented in Section 3.5.

Such a characterization, based on a normal approximation for the density of the estimator, may prove very approximate for a non-LP model obtained from a small number of observations. A more precise (sometimes exact) characterization of density will thus be considered in Section 5.3.3.

5.3.1 Non-Bayesian estimators

Let $\hat{\mathbf{p}}(.)$ be an (absolutely) *unbiased* estimator of $\mathbf{p}^*$, *i.e.* such that

$$\underset{\mathbf{y}^s|\mathbf{p}^*}{\mathrm{E}}\ \hat{\mathbf{p}}(\mathbf{y}^s) = \mathbf{p}^*,$$

which amounts to saying that if it were possible to replicate the same experiment and estimate $\hat{\mathbf{p}}$ an infinite number of times, the mean of the estimates would coincide with the true value.

Let $\mathbf{P}$ be the covariance matrix of this estimator. Since $\hat{\mathbf{p}}(.)$ is unbiased, $\mathbf{P}$ can be written as

$$\mathbf{P} = \underset{\mathbf{y}^s|\mathbf{p}^*}{\mathrm{E}}\ \{[\hat{\mathbf{p}}(\mathbf{y}^s) - \mathbf{p}^*][\hat{\mathbf{p}}(\mathbf{y}^s) - \mathbf{p}^*]^{\mathrm{T}}\},$$

which quantifies how the estimates are spread around the true value $\mathbf{p}^*$. One would like the estimates to be as concentrated as possible around this true value, of course. An estimator $\hat{\mathbf{p}}_1(.)$ with covariance matrix $\mathbf{P}_1$ is said to be *more efficient* than an estimator $\hat{\mathbf{p}}_2(.)$ with covariance matrix $\mathbf{P}_2$ if $\mathbf{P}_1 < \mathbf{P}_2$, that is if $\mathbf{P}_2 - \mathbf{P}_1$ is positive-definite (*i.e.* if all the eigenvalues of $\mathbf{P}_2 - \mathbf{P}_1$ are strictly positive). Since estimators with high efficiency are desirable, a natural request is to make $\mathbf{P}$ as small as possible. The Cramér-Rao inequality provides a lower bound to what can be achieved.

REMARK 5.2

The maximum-likelihood estimator is *asymptotically* unbiased (Section 3.3.3), but generally biased for a finite number of data points. For normal additive noise at the output, an approximation to the bias can be found in (Box, 1971). Several methods have been suggested to reduce this bias (Picard and Prum, 1992; Firth, 1993; Pronzato and Pázman, 1994a). ◊

5.3.1.1 Cramér-Rao inequality

Under the hypotheses that:

— the set of all data vectors $\mathbf{y}^s$ with $\pi_y(\mathbf{y}^s|\mathbf{p}) > 0$ does not depend on $\mathbf{p}$,
— $\partial \ln \pi_y(\mathbf{y}^s|\mathbf{p})/\partial p_i$ $(i = 1, \ldots, n_p)$ is absolutely integrable,
— $\mathrm{E}_{\mathbf{y}^s|\mathbf{p}} \{[\partial \ln \pi_y(\mathbf{y}^s|\mathbf{p})/\partial \mathbf{p}][\partial \ln \pi_y(\mathbf{y}^s|\mathbf{p})/\partial \mathbf{p}]^{\mathrm{T}}\}$ exists and is invertible,

the covariance of any absolutely unbiased estimator satisfies (Fourgeaud and Fuchs, 1967; Goodwin and Payne, 1977; Sorenson, 1980)

$$\mathbf{P} \geq \mathbf{F}^{-1}(\mathbf{p}^*),$$

where $\mathbf{F}$ is the Fisher information matrix, already introduced in Section 3.3.3, given by

$$\mathbf{F}(\mathbf{p}) = \underset{\mathbf{y}^s|\mathbf{p}}{\mathrm{E}} \{[\frac{\partial}{\partial \mathbf{p}} \ln \pi_y(\mathbf{y}^s|\mathbf{p})][\frac{\partial}{\partial \mathbf{p}} \ln \pi_y(\mathbf{y}^s|\mathbf{p})]^{\mathrm{T}}\} = - \underset{\mathbf{y}^s|\mathbf{p}}{\mathrm{E}} \{\frac{\partial^2}{\partial \mathbf{p} \partial \mathbf{p}^{\mathrm{T}}} \ln \pi_y(\mathbf{y}^s|\mathbf{p})\}.$$

Remember that $\ln \pi_y(\mathbf{y}^s|\mathbf{p})$ is the log-likelihood of the data $\mathbf{y}^s$ (Section 3.3), so the gradient of the log-likelihood with respect to the parameters (or its Hessian) is a basic ingredient in the calculation of $\mathbf{F}(\mathbf{p})$.

An estimator that reaches the Cramér-Rao lower bound is said to be *efficient*. Under conditions stated in Section 3.3.3, the maximum-likelihood estimator is asymptotically efficient. In some special cases, this property is also valid for a finite set of data, as will be seen in the next section.

5.3.1.2 LP model structure and normal noise with known covariance

Assume that the data satisfy

$$\mathbf{y}^s = \mathbf{R}\mathbf{p}^* + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon}$ is normally distributed, with zero mean and *known* covariance $\boldsymbol{\Sigma}$. The likelihood of the observations can then be written as

$$\pi_y(\mathbf{y}^s|\mathbf{p}) = [(2\pi)^{n_t} \det \boldsymbol{\Sigma}]^{-1/2} \exp [-\frac{1}{2} (\mathbf{y}^s - \mathbf{R}\mathbf{p})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y}^s - \mathbf{R}\mathbf{p})].$$

The gradient of the log-likelihood is therefore

$$\frac{\partial}{\partial \mathbf{p}} \ln \pi_y(\mathbf{y}^s|\mathbf{p}) = \mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y}^s - \mathbf{R}\mathbf{p}),$$

and the Hessian is

$$\frac{\partial^2}{\partial \mathbf{p} \partial \mathbf{p}^{\mathrm{T}}} \ln \pi_y(\mathbf{y}^s|\mathbf{p}) = -\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R}.$$

Since the Hessian does not depend on $\mathbf{y}^s$, the Fisher information matrix is given by

$$\mathbf{F} = \mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R}.$$

F therefore does not depend on the value of the parameters.

Let us show that, under the same hypotheses, the least-squares estimator, weighted by the inverse of the noise covariance (*i.e.* $\mathbf{Q} = \boldsymbol{\Sigma}^{-1}$),

$$\hat{\mathbf{p}}_{\mathrm{ls}} = (\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{y}^{\mathrm{s}},$$

is unbiased and efficient. Since

$$\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}\{\hat{\mathbf{p}}_{\mathrm{ls}}(\mathbf{y}^{\mathrm{s}})\} = \mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}\{(\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{R}\mathbf{p}^* + \boldsymbol{\varepsilon})\}$$

$$= \mathbf{p}^* + (\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}\{\boldsymbol{\varepsilon}\} = \mathbf{p}^*,$$

$\hat{\mathbf{p}}_{\mathrm{ls}}$ is unbiased. Its covariance matrix is

$$\mathbf{P}_{\mathrm{ls}} = \mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}\{[\hat{\mathbf{p}}_{\mathrm{ls}}(\mathbf{y}^{\mathrm{s}}) - \mathbf{p}^*][\hat{\mathbf{p}}_{\mathrm{ls}}(\mathbf{y}^{\mathrm{s}}) - \mathbf{p}^*]^{\mathrm{T}}\}$$

$$= \mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}\{[(\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{R}\mathbf{p}^*+\boldsymbol{\varepsilon}) - \mathbf{p}^*][(\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{R}\mathbf{p}^*+\boldsymbol{\varepsilon}) - \mathbf{p}^*]^{\mathrm{T}}\}$$

$$= (\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}\{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^{\mathrm{T}}\}\boldsymbol{\Sigma}^{-1}\mathbf{R}(\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1}.$$

Since $\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}^*}\{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^{\mathrm{T}}\} = \boldsymbol{\Sigma}$, this simplifies to

$$\mathbf{P}_{\mathrm{ls}} = (\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1} = \mathbf{F}^{-1},$$

so $\hat{\mathbf{p}}_{\mathrm{ls}}$ is efficient. Note that when calculating $\hat{\mathbf{p}}_{\mathrm{ls}}$ through the explicit formula derived in Section 4.1.3

$$\hat{\mathbf{p}}_{\mathrm{ls}} = (\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{y}^{\mathrm{s}},$$

or through its recursive counterpart, one obtains $\mathbf{P}_{\mathrm{ls}}$ without any additional effort, hence the remarks in Chapter 4 about the interest of knowing the value of $(\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1}$, or, at least, of its diagonal terms. As mentioned in Section 4.1.3.2, it is numerically preferable to use singular-value decomposition for **R**, expressing it as $\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}}$.

Since the model structure is LP, the expectation surface is flat. Consider the projector

$$\boldsymbol{\Pi} = \mathbf{R}(\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1},$$

which coincides with that indicated in Section 5.1.1.1 when $\boldsymbol{\Sigma} = \mathbf{I}_{n_t}\sigma^2$. A $100(1 - \alpha)\%$ confidence region for the parameters is now

$$\mathbb{R}^2_{\alpha} = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid (\mathbf{y}^\mathrm{s} - \mathbf{R}\mathbf{p})^\mathrm{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Pi}(\mathbf{y}^\mathrm{s} - \mathbf{R}\mathbf{p}) \leq \chi^2_{\alpha}(n_\mathrm{p})\}.$$

Since $\mathbf{y}^\mathrm{s} - \mathbf{R}\mathbf{p} = \mathbf{y}^\mathrm{s} - \mathbf{R}\hat{\mathbf{p}}_\mathrm{ls} + \mathbf{R}(\hat{\mathbf{p}}_\mathrm{ls} - \mathbf{p}) = (\mathbf{I}_{n_\mathrm{t}} - \boldsymbol{\Pi})\mathbf{y}^\mathrm{s} + \mathbf{R}(\hat{\mathbf{p}}_\mathrm{ls} - \mathbf{p})$, one can also write

$$\mathbb{R}^2_{\alpha} = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid (\mathbf{p} - \hat{\mathbf{p}}_\mathrm{ls})^\mathrm{T}\mathbf{R}^\mathrm{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Pi}\mathbf{R}(\mathbf{p} - \hat{\mathbf{p}}_\mathrm{ls}) \leq \chi^2_{\alpha}(n_\mathrm{p})\}$$

$$= \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid (\mathbf{p} - \hat{\mathbf{p}}_\mathrm{ls})^\mathrm{T}\mathbf{R}^\mathrm{T}\boldsymbol{\Sigma}^{-1}\mathbf{R}(\mathbf{p} - \hat{\mathbf{p}}_\mathrm{ls}) \leq \chi^2_{\alpha}(n_\mathrm{p})\}$$

$$= \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid (\mathbf{p} - \hat{\mathbf{p}}_\mathrm{ls})^\mathrm{T}\mathbf{F}(\mathbf{p} - \hat{\mathbf{p}}_\mathrm{ls}) \leq \chi^2_{\alpha}(n_\mathrm{p})\},$$

which corresponds to a $100(1-\alpha)\%$ confidence ellipsoid for $\mathbf{p}$. For each parameter p_i, a $100(1-\alpha)\%$ confidence interval can be defined from the distribution of $(\hat{\mathbf{p}}_\mathrm{ls})_i$, which is $\mathcal{N}((\mathbf{p}^*)_i, (\mathbf{F}^{-1})_{ii})$. For instance, a 95% confidence interval is

$$[(\hat{\mathbf{p}}_\mathrm{ls})_i - 2\rho_i, (\hat{\mathbf{p}}_\mathrm{ls})_i + 2\rho_i],$$

with ρ_i the square root of the ith diagonal entry of $\mathbf{F}^{-1}$. It is also interesting to indicate the *correlation coefficient* between the estimated parameters p_i and p_k ($i, k = 1, \ldots, n_\mathrm{p}$), given by

$$-1 \leq c_{ik} = \frac{[\mathbf{F}^{-1}]_{ik}}{[\mathbf{F}^{-1}]^{1/2}_{ii}[\mathbf{F}^{-1}]^{1/2}_{kk}} \leq 1.$$

The results might conveniently be presented as a table where the ith line contains $(\hat{\mathbf{p}}_\mathrm{ls})_i$, the 95% confidence interval $[(\hat{\mathbf{p}}_\mathrm{ls})_i - 2\rho_i, (\hat{\mathbf{p}}_\mathrm{ls})_i + 2\rho_i]$ and the correlation coefficients c_{ik} ($k = 1, \ldots, n_\mathrm{p}$).

REMARK 5.3

The fact that the least-squares estimator weighted by the inverse of the noise covariance is unbiased and efficient does not imply that it minimizes the mean-square error in the parameters. It may sometimes be preferable to accept some bias to reduce the covariance of the estimator (see, *e.g.*, (Norton, 1986a), pp. 109-111). For instance, one can use a *ridge estimate* (Hoerl and Kennard, 1970; Marquardt, 1970; Goldstein and Smith, 1974)

$$\hat{\mathbf{p}}(\mu) = (\mathbf{R}^\mathrm{T}\boldsymbol{\Sigma}^{-1}\mathbf{R} + \mu\mathbf{I}_{n_\mathrm{p}})^{-1}\mathbf{R}^\mathrm{T}\boldsymbol{\Sigma}^{-1}\mathbf{y}^\mathrm{s},$$

the covariance matrix of which is

$$(\mathbf{R}^\mathrm{T}\boldsymbol{\Sigma}^{-1}\mathbf{R} + \mu\mathbf{I}_{n_\mathrm{p}})^{-1}\mathbf{R}^\mathrm{T}\boldsymbol{\Sigma}^{-1}\mathbf{R}(\mathbf{R}^\mathrm{T}\boldsymbol{\Sigma}^{-1}\mathbf{R} + \mu\mathbf{I}_{n_\mathrm{p}})^{-1}.$$

Note the similarity to the Levenberg-Marquardt method for least-squares optimization (Section 4.3.3.5).

The hyperparameter $\mu \geq 0$ must be tuned (Section 3.8) to ensure a satisfactory compromise. When μ tends to zero, the estimator is unbiased and efficient. When μ tends to $+\infty$, $\hat{\mathbf{p}}(\mu)$ tends to $\mathbf{0}$. The ridge estimator thus tends to reduce the variance by favouring the origin of parameter space. ◊

5.3.1.3 LP model structure and normal stationary noise with unknown variance

Assuming that the data satisfy

$$y(t) = \mathbf{r}^{\mathrm{T}}(t)\mathbf{p}^{*} + \varepsilon(t),\ t = 1, \ldots, n_{\mathrm{t}},$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, \sigma^2)$, with σ^2 unknown, amounts to assuming that

$$\mathbf{y}^{\mathrm{s}} = \mathbf{R}\mathbf{p}^{*} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon}$ is distributed $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$. In this case, $\hat{\mathbf{p}}_{\mathrm{ml}}$ is obtained by unweighted least squares ($\mathbf{Q} = \mathbf{I}$, $\hat{\mathbf{p}}_{\mathrm{ml}} = \hat{\mathbf{p}}_{\mathrm{ls}}$), and $\hat{\sigma}^2_{\mathrm{ml}}$ by

$$\hat{\sigma}^2_{\mathrm{ml}} = \frac{1}{n_{\mathrm{t}}}\sum_{t=1}^{n_{\mathrm{t}}} [y(t) - \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}_{\mathrm{ls}}]^2.$$

A first approach would then be to compute the inverse of the Fisher information matrix associated with the *extended* parameter vector

$$\mathbf{p}_{\mathrm{e}} = \begin{bmatrix} \mathbf{p} \\ \sigma^2 \end{bmatrix}.$$

However, it is often enough to act as if σ^2 were known, which amounts to taking the covariance matrix of $\hat{\mathbf{p}}_{\mathrm{ls}}$ as

$$\mathbf{P}_{\mathrm{ls}} = \sigma^2(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1},$$

and, since the value of σ^2 is unknown, replacing it by an estimate. As $\hat{\sigma}^2_{\mathrm{ml}}$ is biased (it is only asymptotically unbiased), the unbiased estimator

$$\hat{\sigma}^2 = \frac{1}{n_{\mathrm{t}} - n_{\mathrm{p}}}\sum_{t=1}^{n_{\mathrm{t}}} [y(t) - \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}_{\mathrm{ls}}]^2$$

is usually preferred. The confidence region $\mathbb{R}_3^{\alpha}$ then becomes

$$\mathbb{R}_3^{\alpha} = \{\mathbf{p} \in \mathbb{R}^{n_{\mathrm{p}}} \mid \frac{(\mathbf{y}^{\mathrm{s}} - \mathbf{R}\mathbf{p})^{\mathrm{T}}\boldsymbol{\Pi}(\mathbf{y}^{\mathrm{s}} - \mathbf{R}\mathbf{p})}{n_{\mathrm{p}}\hat{\sigma}^2} \leq F_{\alpha}(n_{\mathrm{p}}, n_{\mathrm{t}} - n_{\mathrm{p}})\}.$$

Since

$$\boldsymbol{\Pi} = \mathbf{R}(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}},$$

$\mathbb{R}_3^{\alpha}$ can also be rewritten as

$$\mathbb{R}_3^{\alpha} = \{\mathbf{p} \in \mathbb{R}^{n_{\mathrm{p}}} \mid \frac{(\mathbf{p} - \hat{\mathbf{p}}_{\mathrm{ls}})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{R}(\mathbf{p} - \hat{\mathbf{p}}_{\mathrm{ls}})}{n_{\mathrm{p}}\hat{\sigma}^2} \leq F_{\alpha}(n_{\mathrm{p}}, n_{\mathrm{t}} - n_{\mathrm{p}})\},$$

which permits construction of exact confidence ellipsoids (Section 5.1.1.2). Moreover,

$$\frac{(\mathbf{p}^*)_i - (\hat{\mathbf{p}}_{\mathrm{ls}})_i}{\hat{\sigma}[(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}]_{ii}^{1/2}}, \quad i = 1, \ldots, n_{\mathrm{p}},$$

has a Student t-distribution with $n_{\mathrm{t}} - n_{\mathrm{p}}$ degrees of freedom, which can be used to derive exact $100(1 - \alpha)\%$ confidence intervals for the parameters.

5.3.1.4 Other cases

Under the hypotheses mentioned in Section 3.3.3, the maximum-likelihood estimator is *asymptotically* normal and *asymptotically* efficient. The estimator $\hat{\mathbf{p}}_{\mathrm{ml}}$ thus tends to be distributed $\mathcal{N}(\mathbf{p}^*, \mathbf{F}^{-1}(\mathbf{p}^*))$ as the number of data points tends to infinity, hence the idea of characterizing the uncertainty in $\hat{\mathbf{p}}_{\mathrm{ml}}$ by $\mathbf{F}^{-1}(\hat{\mathbf{p}}_{\mathrm{ml}})$. This relies on a chain of approximations, and the result should be viewed with some scepticism. Assume, for instance, that the data satisfy

$$y(t_i) = y_{\mathrm{m}}(t_i, \mathbf{p}^*) + \varepsilon(t_i), \quad i = 1, \ldots, n_{\mathrm{t}},$$

where the $\varepsilon(t_i)$'s are independent random variables distributed $\mathcal{N}(0, \sigma_{t_i}^2)$, with $\sigma_{t_i}^2$ known. The associated log-likelihood can be written as

$$\ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}) = (\text{term independent of } \mathbf{p}) - \frac{1}{2} \sum_{i=1}^{n_{\mathrm{t}}} \frac{[y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})]^2}{\sigma_{t_i}^2}.$$

Its gradient is thus

$$\frac{\partial}{\partial \mathbf{p}} \ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p}) = \sum_{i=1}^{n_{\mathrm{t}}} \frac{1}{\sigma_{t_i}^2} [y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})] \frac{\partial}{\partial \mathbf{p}} y_{\mathrm{m}}(t_i, \mathbf{p}).$$

The Fisher information matrix can now be calculated as

$$\begin{aligned}\mathbf{F}(\mathbf{p}) &= \mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}} \{ [\frac{\partial}{\partial \mathbf{p}} \ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p})][\frac{\partial}{\partial \mathbf{p}} \ln \pi_{\mathrm{y}}(\mathbf{y}^{\mathrm{s}}|\mathbf{p})]^{\mathrm{T}} \} \\ &= \mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}} \Big\{ \sum_{k=1}^{n_{\mathrm{t}}} \frac{1}{\sigma_{t_k}^2} \frac{\partial y_{\mathrm{m}}(t_k, \mathbf{p})}{\partial \mathbf{p}} [y(t_k) - y_{\mathrm{m}}(t_k, \mathbf{p})] \\ &\quad \times \sum_{i=1}^{n_{\mathrm{t}}} [y(t_i) - y_{\mathrm{m}}(t_i, \mathbf{p})] \frac{1}{\sigma_{t_i}^2} \frac{\partial y_{\mathrm{m}}(t_i, \mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}} \Big\}.\end{aligned}$$

Since

$$\underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}}{\mathrm{E}}\{[y(t_k)-y_{\mathrm{m}}(t_k,\mathbf{p})][y(t_i)-y_{\mathrm{m}}(t_i,\mathbf{p})]\} = \sigma_{t_i}^2\delta_{ik},$$

one gets

$$\mathbf{F}(\mathbf{p}) = \sum_{i=1}^{n_t} \frac{1}{\sigma_{t_i}^2} \frac{\partial y_{\mathrm{m}}(t_i,\mathbf{p})}{\partial \mathbf{p}} \frac{\partial y_{\mathrm{m}}(t_i,\mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}} .$$

$\mathbf{F}(\mathbf{p})$ is therefore the approximation of the Hessian used in the Gauss-Newton algorithm (Sections 4.2.4 and 4.3.3.4), hence the interest, already mentioned in Chapter 4, of inspecting the value of the inverse of the Hessian once the algorithm has converged.

Using $\mathbf{F}^{-1}(\hat{\mathbf{p}}_{\mathrm{ml}})$ to characterize the uncertainty in the parameters relies on the following chain of approximations:

A1: $\mathbf{F}^{-1}(\mathbf{p}^*)$ is substituted for $\mathbf{P}_{\mathrm{ml}}$; however, the number of observations is always finite, and sometimes quite small, so $\hat{\mathbf{p}}_{\mathrm{ml}}$ is generally biased.

A2: $\mathbf{F}^{-1}(\hat{\mathbf{p}}_{\mathrm{ml}})$ is substituted for $\mathbf{F}^{-1}(\mathbf{p}^*)$.

A3: $\sum_{i=1}^{n_t} \frac{1}{\sigma_{t_i}^2} \frac{\partial y_{\mathrm{m}}(t_i,\mathbf{p})}{\partial \mathbf{p}} \frac{\partial y_{\mathrm{m}}(t_i,\mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}$ is substituted for $\mathbf{F}(\mathbf{p})$, whereas the hypotheses on the noise which allow this expression are never totally satisfied.

When compared to the approaches presented in Sections 5.1 and 5.2, this has the advantage of requiring far less computation, since the estimate of $\mathbf{P}_{\mathrm{ml}}$ is obtained as a by-product of the optimization procedure. This explains why it is certainly the most commonly used, although one should be suspicious of its results, because of all the approximations involved. These results are more credible insofar as

— the number of data points is large,
— the nonlinearity of the model with respect to its parameters is mild,
— the measurement errors are independently distributed and have small magnitudes.

Generally it is thought enough to give, together with the estimated value of the ith parameter p_i, the square root ρ_i of the ith diagonal element of $\mathbf{F}^{-1}(\hat{\mathbf{p}}_{\mathrm{ml}})$, which forms an estimate of the associated standard deviation. One thus obtains an *approximate* 95% confidence interval for the ith parameter in the form $[\hat{p}_{i_{\mathrm{ml}}} - 2\rho_i, \hat{p}_{i_{\mathrm{ml}}} + 2\rho_i]$. An *approximate* correlation coefficient between the ith and kth estimated parameters $(i, k = 1, \dots, n_{\mathrm{p}})$ is given by

$$-1 \leq c_{ik} = \frac{[\mathbf{F}^{-1}(\hat{\mathbf{p}}_{\mathrm{ml}})]_{ik}}{[\mathbf{F}^{-1}(\hat{\mathbf{p}}_{\mathrm{ml}})]_{ii}^{1/2}[\mathbf{F}^{-1}(\hat{\mathbf{p}}_{\mathrm{ml}})]_{kk}^{1/2}} \leq 1 .$$

REMARKS 5.4

— If the $\sigma_{t_i}^2$'s were unknown, the approach advocated in Section 3.3.1, Example 3.3, would lead to estimating the extended parameter vector

$$\mathbf{p}_{\mathrm{e}} = \begin{bmatrix} \mathbf{p} \\ \mathbf{p}_{\sigma} \end{bmatrix}$$

by maximum likelihood, with $\mathbf{p}_{\sigma}$ the parameters of the noise variance, then calculating $\mathbf{F}^{-1}(\hat{\mathbf{p}}_{\mathrm{eml}})$. In practice, as in Section 5.3.1.3, one is often content to act as if each noise variance $\sigma_{t_i}^2$ were equal to its estimate $\hat{\sigma}_{t_i}^2$, which amounts to taking the Fisher information matrix as

$$\mathbf{F}(\mathbf{p}) = \sum_{i=1}^{n_t} \frac{1}{\hat{\sigma}_{t_i}^2} \frac{\partial y_{\mathrm{m}}(t_i, \mathbf{p})}{\partial \mathbf{p}} \frac{\partial y_{\mathrm{m}}(t_i, \mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}},$$

In the special case where all $\sigma_{t_i}^2$'s are equal, $\hat{\mathbf{p}}_{\mathrm{ml}}$ is estimated by unweighted least squares and σ^2 by

$$\hat{\sigma}^2 = \frac{1}{n_t - n_{\mathrm{p}}} \sum_{i=1}^{n_t} [y(t_i) - y_{\mathrm{m}}(t_i, \hat{\mathbf{p}}_{\mathrm{ml}})]^2.$$

$\mathbf{F}(\mathbf{p}^*)$ is then approximated by

$$\mathbf{F}(\hat{\mathbf{p}}_{\mathrm{ml}}) = \frac{1}{\hat{\sigma}^2} \sum_{i=1}^{n_t} \frac{\partial y_{\mathrm{m}}(t_i, \mathbf{p})}{\partial \mathbf{p}} \frac{\partial y_{\mathrm{m}}(t_i, \mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}} \bigg|_{\mathbf{p} = \hat{\mathbf{p}}_{\mathrm{ml}}},$$

and the set

$$\{\mathbf{p} \in \mathbb{R}^{n_{\mathrm{p}}} \mid \frac{(\mathbf{p} - \hat{\mathbf{p}}_{\mathrm{ml}})^{\mathrm{T}} \mathbf{F}(\hat{\mathbf{p}}_{\mathrm{ml}})(\mathbf{p} - \hat{\mathbf{p}}_{\mathrm{ml}})}{n_{\mathrm{p}}} \leq F_{\alpha}(n_{\mathrm{p}}, n_t - n_{\mathrm{p}})\}$$

defines an approximate $100(1 - \alpha)\%$ confidence ellipsoid for $\mathbf{p}$. As already mentioned, $\mathbf{F}(\hat{\mathbf{p}}_{\mathrm{ml}})$ is equal, up to multiplication by a scalar, to the approximate Hessian used in the Gauss-Newton algorithm. The ratio of the largest to the smallest eigenvalue of $\mathbf{F}(\hat{\mathbf{p}}_{\mathrm{ml}})$ is thus clearly related to the numerical conditioning of the optimization problem.

— For each parameter p_i, an approximate confidence interval can be obtained by using the fact that

$$\frac{(\mathbf{p}^*)_i - (\hat{\mathbf{p}}_{\mathrm{ml}})_i}{[\mathbf{F}^{-1}(\hat{\mathbf{p}}_{\mathrm{ml}})]_{ii}^{1/2}}$$

approximately follows a Student t-distribution with $n_t - n_{\mathrm{p}}$ degrees of freedom.

— When the $\varepsilon(t_i)$'s correspond to independent random variables with non-Gaussian densities $\pi_{\varepsilon}(\varepsilon, t_i)$, it is easy to show that the expression for the Fisher information matrix is similar to the Gaussian case, with $1/\sigma_{t_i}^2$ simply replaced by the Fisher information

$$I(t_i) = \int_{\mathbb{D}} \left(\frac{\partial \ln \pi_\varepsilon(\varepsilon, t_i)}{\partial \varepsilon}\right)^2 \pi_\varepsilon(\varepsilon, t_i)\, d\varepsilon,$$

assumed to exist, with $\mathbb{D} = \{\varepsilon \mid \pi_\varepsilon(\varepsilon, t_i) > 0\}$ (see also Section 3.7.1).

— If the output errors for the true value of the parameters do not correspond to independent random variables, prediction errors should be considered instead, as in Section 3.3.2 for maximum-likelihood estimation. In general, the Fisher information matrix can then no longer be written as a sum of rank-one matrices, each associated with a single observation; see Section 6.3.2.2. ◊

5.3.2 Bayesian estimators

As already mentioned in Section 3.5.1, the maximum *a posteriori* estimator has the same asymptotic properties of consistency and efficiency as the maximum-likelihood estimator $\hat{\mathbf{p}}_{\mathrm{ml}}(\mathbf{y}^{\mathrm{s}})$, provided $\pi_{\mathrm{p}}(\mathbf{p})$ is continuous and non-zero at $\hat{\mathbf{p}}_{\mathrm{ml}}(\mathbf{y}^{\mathrm{s}})$. The asymptotic uncertainty in $\hat{\mathbf{p}}_{\mathrm{map}}(\mathbf{y}^{\mathrm{s}})$ can therefore also be characterized by the Fisher information matrix, since the prior information asymptotically vanishes. However, this prior information should still be taken into account when the number of observations is finite.

The Cramér-Rao inequality can be extended to the Bayesian estimators considered in Section 3.5. In this context, an estimator $\hat{\mathbf{p}}(\mathbf{y}^{\mathrm{s}})$ is said to be unbiased if

$$\mathop{\mathrm{E}}_{\mathbf{p}}\{\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}} \hat{\mathbf{p}}(\mathbf{y}^{\mathrm{s}})\} = \mathop{\mathrm{E}}_{\mathbf{p}}\{\mathbf{p}\},$$

where $\mathbf{p}$ has a prior distribution $\pi_{\mathrm{p}}(\mathbf{p})$. Under conditions similar to those required for the Cramér-Rao inequality to be valid, the covariance matrix of the estimation error $\hat{\mathbf{p}}(\mathbf{y}^{\mathrm{s}}) - \mathbf{p}$ satisfies, for any unbiased estimator $\hat{\mathbf{p}}(\mathbf{y}^{\mathrm{s}})$ (see, *e.g.*, (Sorenson, 1980))

$$\mathop{\mathrm{E}}_{\mathbf{p}}\{\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}}\{[\hat{\mathbf{p}}(\mathbf{y}^{\mathrm{s}}) - \mathbf{p}][\hat{\mathbf{p}}(\mathbf{y}^{\mathrm{s}}) - \mathbf{p}]^{\mathrm{T}}\}\} \geq$$

$$\Big[\mathop{\mathrm{E}}_{\mathbf{p}}\{\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}}\{[\frac{\partial}{\partial \mathbf{p}} \ln \pi(\mathbf{y}^{\mathrm{s}}, \mathbf{p})][\frac{\partial}{\partial \mathbf{p}} \ln \pi(\mathbf{y}^{\mathrm{s}}, \mathbf{p})]^{\mathrm{T}}\}\}\Big]^{-1}$$

$$= \Big[\mathop{\mathrm{E}}_{\mathbf{p}}\{\mathbf{F}(\mathbf{p})\} + \mathop{\mathrm{E}}_{\mathbf{p}}\{[\frac{\partial}{\partial \mathbf{p}} \ln \pi_{\mathrm{p}}(\mathbf{p})][\frac{\partial}{\partial \mathbf{p}} \ln \pi_{\mathrm{p}}(\mathbf{p})]^{\mathrm{T}}\}\Big]^{-1},$$

where $\mathbf{F}(\mathbf{p})$ is the Fisher information matrix.

We have seen in Section 3.5.2 that the minimum-risk estimator for a quadratic cost coincides with the posterior mean of $\mathbf{p}$, which is unbiased. The covariance matrix of the associated estimation error thus satisfies the inequality above. However, this estimator is not absolutely unbiased, so the Cramér-Rao inequality of Section 5.3.1.1 does not apply. Consider the special case of an LP model structure

$$\mathbf{y}^{\mathrm{s}} = \mathbf{R}\mathbf{p}^* + \boldsymbol{\varepsilon},$$

with $\boldsymbol{\varepsilon}$ distributed $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ and $\boldsymbol{\Sigma}$ known. We have already shown that

$$\mathbf{F}(\mathbf{p}) = \mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R}.$$

Assume, moreover, that the prior density $\pi_{\mathrm{p}}(\mathbf{p})$ is normal $\mathcal{N}(\mathbf{p}_0, \boldsymbol{\Omega})$. One can then easily check that

$$\underset{\mathbf{p}}{\mathrm{E}}\,\{\mathbf{F}(\mathbf{p})\} + \underset{\mathbf{p}}{\mathrm{E}}\,\{[\frac{\partial}{\partial\mathbf{p}}\ln \pi_{\mathrm{p}}(\mathbf{p})][\frac{\partial}{\partial\mathbf{p}}\ln \pi_{\mathrm{p}}(\mathbf{p})]^{\mathrm{T}}\} = \mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R} + \boldsymbol{\Omega}^{-1},$$

and that the bound of the inequality is reached for the maximum *a posteriori* estimator $\hat{\mathbf{p}}_{\mathrm{map}}$, so

$$\underset{\mathbf{p}}{\mathrm{E}}\,\{\underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}}{\mathrm{E}}\,\{[\hat{\mathbf{p}}_{\mathrm{map}}(\mathbf{y}^{\mathrm{s}}) - \mathbf{p}][\hat{\mathbf{p}}_{\mathrm{map}}(\mathbf{y}^{\mathrm{s}}) - \mathbf{p}]^{\mathrm{T}}\}\} = (\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R} + \boldsymbol{\Omega}^{-1})^{-1}.$$

REMARK 5.5

The estimator $\hat{\mathbf{p}}_{\mathrm{map}}(\mathbf{y}^{\mathrm{s}})$ also minimizes the risk

$$j_{\mathrm{mr}}(\mathbf{p}) = \int(\mathbf{p} - \mathbf{p}^{*})^{\mathrm{T}}\mathbf{K}^{\mathrm{T}}\mathbf{K}(\mathbf{p} - \mathbf{p}^{*})\pi_{\mathrm{p}}(\mathbf{p}^{*}|\mathbf{y}^{\mathrm{s}})d\mathbf{p}^{*},$$

where $\mathbf{K}$ is any matrix with rank dim $\mathbf{p}$ (Section 3.5.2). The prior expectation of the minimum risk is then

$$\underset{\mathbf{y}^{\mathrm{s}}}{\mathrm{E}}\,\{j_{\mathrm{mr}}[\hat{\mathbf{p}}_{\mathrm{map}}(\mathbf{y}^{\mathrm{s}})]\} = \mathrm{trace}\,[\mathbf{K}(\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R} + \boldsymbol{\Omega}^{-1})^{-1}\mathbf{K}^{\mathrm{T}}].$$

It can be chosen as a cost function to select the experimental conditions; see Section 6.5. The matrix $(\mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R} + \boldsymbol{\Omega}^{-1})/n_{\mathrm{t}}$ is called the *Bayesian information matrix* (Pilz, 1983). ◊

5.3.3 Approximation of the probability density of the estimator

The probability density of the estimator in a sense summarizes the information about the precision of the estimation. Let $\mathbf{p}^{*}$ and Ξ respectively denote the vectors of the true values of the parameters and of the experimental conditions used to obtain the n_{t} observations (Chapter 6). An approximation $q_{\Xi}(\hat{\mathbf{p}}_{\mathrm{ls}}|\mathbf{p}^{*})$ of the density of the least-squares estimator in the case of additive noise distributed $\mathcal{N}(0, \sigma^2)$ has been obtained by Pázman (1984, 1990, 1993) as

$$q_\Xi(\hat{\mathbf{p}}_{\mathrm{ls}}|\mathbf{p}^*) = \frac{|\det \mathbf{Q}_\Xi(\hat{\mathbf{p}}_{\mathrm{ls}}, \mathbf{p}^*)|}{(2\pi)^{n_\mathrm{p}/2} \det^{1/2} \mathbf{F}(\hat{\mathbf{p}}_{\mathrm{ls}}, \Xi)}$$

$$\times \exp \{-\frac{1}{2\sigma^2} \| \mathbf{\Pi}_\Xi(\hat{\mathbf{p}}_{\mathrm{ls}})[\mathbf{y}^\mathrm{m}(\Xi, \hat{\mathbf{p}}_{\mathrm{ls}}) - \mathbf{y}^\mathrm{m}(\Xi, \mathbf{p}^*)] \|_2^2 \},$$

where the matrix $\mathbf{\Pi}_\Xi(\hat{\mathbf{p}}_{\mathrm{ls}})$ is the orthogonal projector onto the tangent plane to $\mathbb{S}_{\mathrm{exp}}$ at $\mathbf{y}^\mathrm{m}(\Xi, \hat{\mathbf{p}}_{\mathrm{ls}})$ (Section 5.1.1), and

$$[\mathbf{Q}_\Xi(\hat{\mathbf{p}}_{\mathrm{ls}}, \mathbf{p}^*)]_{i,k} = [\mathbf{F}(\hat{\mathbf{p}}_{\mathrm{ls}}, \Xi)]_{i,k}$$

$$+ \frac{1}{\sigma^2} [\mathbf{y}^\mathrm{m}(\Xi, \hat{\mathbf{p}}_{\mathrm{ls}}) - \mathbf{y}^\mathrm{m}(\Xi, \mathbf{p}^*)]^\mathrm{T}[\mathbf{I}_{n_\mathrm{t}} - \mathbf{\Pi}_\Xi(\hat{\mathbf{p}}_{\mathrm{ls}})] \left.\frac{\partial^2 \mathbf{y}^\mathrm{m}(\Xi, \mathbf{p})}{\partial p_i \partial p_k}\right|_{\hat{\mathbf{p}}_{\mathrm{ls}}} .$$

This expression has been obtained by Pázman in 1984 in the non-asymptotic case (n_t finite) through a geometrical approach, and independently by Skovgaard (1985) and Hougaard (1985) as an asymptotic approximation, more precise than the classical normal density. The density $q_\Xi(\hat{\mathbf{p}}_{\mathrm{ls}}|\mathbf{p}^*)$ is exactly that of $\hat{\mathbf{p}}_{\mathrm{ls}}$ when the intrinsic curvature of the model is zero, that is when the expectation surface $\mathbb{S}_{\mathrm{exp}}$ is flat. This will be so, in particular, when the number of distinct experimental conditions in Ξ is n_p. The density is almost exact (in the sense that estimates associated with data $\mathbf{y}^\mathrm{s}$ located at a distance from $\mathbb{S}_{\mathrm{exp}}$ larger than its radius of curvature are neglected) for flat models (*i.e.* those with a Riemannian curvature tensor equal to zero (Amari, 1985)). All models which are functions of a single parameter, or depend nonlinearly on a single parameter, are flat. This is so, for instance, for the Hill model

$$y_\mathrm{m}(\xi, \mathbf{p}) = \frac{\mathrm{E}_{\mathrm{max}}\xi}{\xi_{50} + \xi},$$

with $\mathbf{p} = (\mathrm{E}_{\mathrm{max}}, \xi_{50})^\mathrm{T}$ and ξ_{50} the value of ξ such that $y_\mathrm{m}(\xi_{50}, \mathbf{p}) = \mathrm{E}_{\mathrm{max}}/2$. This is also called the Michaelis-Menten model when used to describe nonlinear effects in biology and pharmacokinetics.

Note that the accuracy of $q_\Xi(\hat{\mathbf{p}}_{\mathrm{ls}}|\mathbf{p}^*)$ depends on the intrinsic nonlinearity of the model (the curvature of $\mathbb{S}_{\mathrm{exp}}$), but not on its parametric nonlinearity. This density may turn out to be quite different from the classical normal approximation, as illustrated by the following example.

EXAMPLE 5.2

Consider the single-parameter model

$$y_\mathrm{m}(\xi, p) = p + p^2 + \xi p^3.$$

Figure 5.10 presents the *exact* density $q_\Xi(\hat{p}_{\mathrm{ls}}|p^*)$ and its normal approximation for $p^* = 0$, $\sigma^2 = 5$ and a single observation at $\xi = 1$. ◊

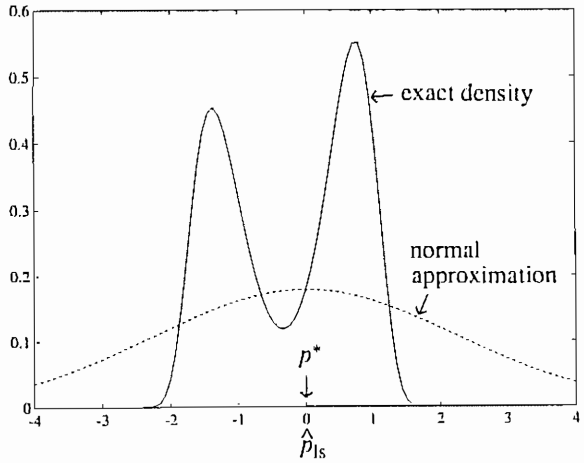


Figure 5.10. Exact density and normal approximation in Example 5.2

The construction of $q_\Xi(\hat{\mathbf{p}}_{ls}|\mathbf{p}^*)$ is *via* the definition of a density on $\mathbb{S}_{\exp}$, so constraints on **p** due to the structure of the model are automatically taken into account, as illustrated by the following example.

EXAMPLE 5.3

Consider the model

$$y_m(\xi, p) = \xi \exp(-\xi) \ln(p\xi).$$

Figure 5.11 shows the density $q_\Xi(\hat{p}_{ls}|p^*)$ and the normal asymptotic density when $p^* = 1$, $\Xi = (1,\ 1)^T$ and $\sigma^2 = 0.25$. $\mathbb{S}_{\exp}$, a straight line with unit slope in the plane $[y(\xi^1), y(\xi^2)]$, is totally described as p varies from zero to infinity, so there cannot be negative least-squares estimates. This is properly taken into account by $q_\Xi(\hat{p}_{ls}|p^*)$ but not by the normal approximation. Since $\mathbb{S}_{\exp}$ is flat, the density $q_\Xi(\hat{p}_{ls}|p^*)$ is exact. ◊

When dim $\mathbf{p} \leq 2$ (as in the previous examples), a plot of $q_\Xi(\mathbf{p}|\mathbf{p}^*)$ as a function of **p**, with the unknown true value $\mathbf{p}^*$ replaced by its estimate $\hat{\mathbf{p}}_{ls}(\mathbf{y}^s)$, allows visual assessment of the precision of the estimation. Since $q_\Xi(\mathbf{p}|\mathbf{p}^*)$ depends on $\mathbf{p}^*$, one should, however, check how it behaves as $\mathbf{p}^*$ varies in the neighbourhood of the estimate $\hat{\mathbf{p}}_{ls}(\mathbf{y}^s)$.

When dim $\mathbf{p} > 2$, marginal densities can be computed. More generally, let $\Gamma[\hat{\mathbf{p}}_{ls}(\mathbf{y}^s)]$ be the function of interest. Its density can be approximated (Pázman and Pronzato, 1994, 1995) by

$$q(\gamma) = \frac{1}{\sigma\sqrt{2\pi}\ \|\mathbf{b}_\gamma\|_2} \exp\{-\frac{1}{2\sigma^2} \|\ \mathbf{P}_\gamma[\mathbf{y}^m(\Xi, \mathbf{p}_\gamma) - \mathbf{y}^m(\Xi, \mathbf{p}^*)]\ \|_2^2\},$$

where

$$\mathbf{p}_\gamma = \arg \min_{\substack{\mathbf{p} \in \mathbb{P} \\ \Gamma(\mathbf{p})=\gamma}} \| \mathbf{y}^{\mathrm{m}}(\Xi, \mathbf{p}) - \mathbf{y}^{\mathrm{m}}(\Xi, \mathbf{p}^*) \|_2^2,$$

$$\mathbf{b}_\gamma = \frac{1}{\sigma^2} \frac{\partial \mathbf{y}^{\mathrm{m}}(\Xi, \mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}\Big|_{\mathbf{p}_\gamma} \mathbf{F}^{-1}(\mathbf{p}_\gamma, \Xi) \frac{\partial \Gamma(\mathbf{p})}{\partial \mathbf{p}}\Big|_{\mathbf{p}_\gamma},$$

and

$$\mathbf{P}_\gamma = \frac{\mathbf{b}_\gamma \mathbf{b}_\gamma^{\mathrm{T}}}{\|\mathbf{b}_\gamma\|_2^2} .$$

The ith marginal density is then simply obtained for $\Gamma(\mathbf{p}) = p_i$. More precise, but more complicated, approximations are suggested in (Pázman and Pronzato, 1994, 1995). The entropy of the density can also be used to quantify the precision of the estimate; see Section 6.4.1, and (Pronzato and Pázman, 1994b) for an approximation to the entropy of the density $q_\Xi(\mathbf{p}|\mathbf{p}^*)$.

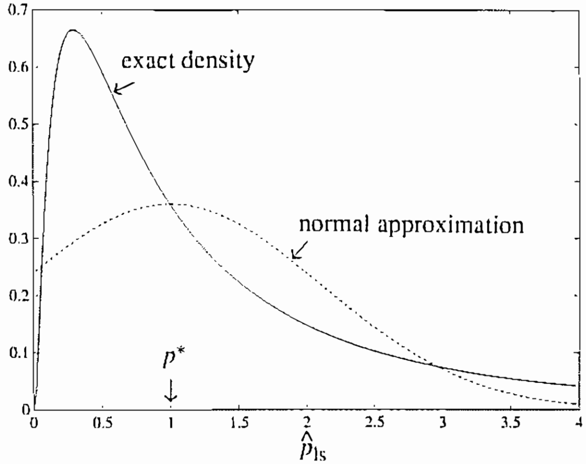


Figure 5.11. Exact density and normal approximation in Example 5.3

An analytical approximation to the density of the Bayesian maximum *a posteriori* estimator will be given in Section 6.4.1, as a particular case of the density of a constrained least-squares estimator, where the constraint is introduced through a penalty function (Pázman and Pronzato, 1992a, 1992b). Its accuracy is the same as for $q_\Xi(\hat{\mathbf{p}}_{\mathrm{ls}}|\mathbf{p}^*)$.

5.4 Bounded-error set estimation

Two kinds of error cannot be avoided when estimating parameters from experimental data. The first corresponds to measurement errors or other perturbations which impose uncertainty on the data. The second corresponds to structural errors, due to the fact that the model structure is at best an approximation of reality. This problem will be considered again in Section 6.6. When the parameters to be estimated have a concrete meaning (phenomenological models), or when decisions have to be based on their numerical values (prediction, diagnosis, control...) one should try to evaluate how these two types of error will affect the estimates.

The usual statistical framework assumes, as we have just seen, that the data are corrupted by errors modelled as realizations of independent random variables, with a known or parametrized distribution. The estimator is then obtained by maximum-likelihood techniques (if the hypotheses only concern the prior distribution of errors) or Bayesian techniques (if the hypotheses also concern the prior distribution of the parameters). Most often, the quality of the estimate is characterized by taking advantage of properties of the Fisher information matrix, since this is by far the easiest method computationally. To the limitations of this approach already mentioned, one may add that it is badly adapted to

— *structural deterministic errors*, such as those encountered when a simple linear model is used to describe the behaviour of a complex deterministic nonlinear model; modelling such errors as realizations of random variables is then questionable, given the perfectly repeatable nature of the deviations;

— errors for which the hypothesis of mutual independence is not tenable;

— errors for which the only prior information is in the form of bounds; this will be the case, for example, for data collected through an analogue-to-digital converter or for measurements performed with a sensor of a given type.

The *bounded-error approach* presented in this section aims to characterize the set of all values of $\mathbf{p}$ that are feasible *a posteriori*, in the sense that the associated errors (which may be deterministic or random) lie between given prior bounds. This approach has received growing attention in the recent literature, as illustrated by special issues of *Mathematics and Computers in Simulation* (Walter, 1990) and the *International Journal of Adaptive Control and Signal Processing* (Norton, 1994, 1995), and a recent research monograph (Milanese *et al.*, 1996). An extensive list of references can be found in the surveys (Combettes, 1993; Deller, Nayeri and Odeh, 1993). See also (Kuntzevich and Lychak, 1992). The methodology extends to many problems where a set defined by inequalities is to be characterized (Walter and Pronzato, 1994).

Of course, one may object that the knowledge of prior bounds on admissible errors can be taken into account through the use of uniform probability densities (Example 3.6). However, characterization of the set of all parameter vectors that are maximum-likelihood estimates will then rely on techniques similar to those presented below, so the distinction between the two approaches becomes pointless.

Note that bounded but independent errors can still be treated with a more conventional approach, *e.g.* least squares, with a suitable choice of error variance. An important condition for the least-squares estimator to have nice asymptotic properties in a bounded-error context is that the sample cross-correlation between the inputs and disturbances tends to zero as n_t increases (Hjalmarsson and Ljung, 1994).

Consider an output error, defined as

$$e_y(k, \mathbf{p}) = y(k) - y_m(k, \mathbf{p}), \quad k = 1, \ldots, n_t,$$

where $y(k)$ is the kth scalar measurement from the system, and $y_m(k, \mathbf{p})$ is the corresponding model output. Other types of errors will be considered in Section 5.4.2.1. Assume $\mathbf{p}$ to be feasible if and only if

$$e_{\mathrm{y}}^{\mathrm{m}}(\mathrm{k}) \le e_{\mathrm{y}}(k, \mathbf{p}) \le e_{\mathrm{y}}^{\mathrm{M}}(k), \quad k = 1, \dots, \mathrm{n_t},$$

where the bounds $e_{\mathrm{y}}^{\mathrm{m}}(k)$ and $e_{\mathrm{y}}^{\mathrm{M}}(k)$ are known *a priori* and $e_{\mathrm{y}}^{\mathrm{m}}(k) \neq e_{\mathrm{y}}^{\mathrm{M}}(k)$. These bounds may come from technical specifications provided by sensor manufacturers, empirical knowledge, or simply intuition. We aim to characterize the set of all values of $\mathbf{p}$ yielding feasible errors, and will denote the *posterior feasible set* associated with the first n_t data by $\mathbb{P}_{n_t}$, with $\mathbb{P}_0$ the prior feasible set. In what follows, we shall assume that $\mathbb{P}_0$ is either $\mathbb{R}^{n_\mathrm{p}}$ or a convex polyhedron in $\mathbb{R}^{n_\mathrm{p}}$, which can be defined by a finite number of linear inequalities. $\mathbb{P}_{n_t}$ is also called the *likelihood set* (Example 3.6). Since $e_{\mathrm{y}}^{\mathrm{m}}(k)$ differs from $e_{\mathrm{y}}^{\mathrm{M}}(k)$, the inequalities associated with the kth measurement can be put in the *standard* form:

$$-1 \le \overline{y}(k) - \overline{y}_{\mathrm{m}}(k, \mathbf{p}) \le 1,$$

with

$$\overline{y}(k) = \frac{2y(k) - e_{\mathrm{y}}^{\mathrm{M}}(k) - e_{\mathrm{y}}^{\mathrm{m}}(k)}{e_{\mathrm{y}}^{\mathrm{M}}(k) - e_{\mathrm{y}}^{\mathrm{m}}(k)} \quad \text{and} \quad \overline{y}_{\mathrm{m}}(k, \mathbf{p}) = \frac{2}{e_{\mathrm{y}}^{\mathrm{M}}(k) - e_{\mathrm{y}}^{\mathrm{m}}(k)}\, y_{\mathrm{m}}(k, \mathbf{p}).$$

For notational convenience, we shall assume that this transformation has been performed, but drop the upper bars on $y(k)$ and $y_{\mathrm{m}}(k, \mathbf{p})$. The set of $\mathbf{p}$ values which satisfy the constraints associated with $y(k)$ can then be written as:

$$\mathbb{\Pi}_k = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid -1 \le y(k) - y_{\mathrm{m}}(k, \mathbf{p}) \le 1\},$$

and the set $\mathbb{P}_n$ of all values of $\mathbf{p}$ consistent with the first n data points is given by the intersection of the sets $\mathbb{\Pi}_k$ ($k = 1, \dots, n$, $n \le n_t$) with $\mathbb{P}_0$.

$\mathbb{P}_{n_t}$ is generally not a singleton, so the associated estimator is not a point-estimator. $\mathbb{P}_{n_t}$ may be empty if the hypotheses are wrong. On the other hand, if there exists some $\mathbf{p}^*$ for which the bounds on the errors are satisfied, then $\mathbb{P}_{n_t}$ certainly contains it.

As in Chapter 4, the algorithms used to characterize $\mathbb{P}_{n_t}$ depend on whether the model structure is LP or not (more generally on whether or not the error is affine in the parameters). The LP case is treated in the next section and the non-LP case will be discussed in Section 5.4.2.

5.4.1 LP model structures

Consider an LP model structure, defined by

$$y_{\mathrm{m}}(k{+}1, \mathbf{p}) = \mathbf{r}^{\mathrm{T}}(k)\mathbf{p}$$

and an *output* error (although other situations involving errors affine in $\mathbf{p}$ could be considered as well). Since the regressor vector $\mathbf{r}(k)$ is assumed to be known, $\mathbb{\Pi}_k$ is a strip in parameter space, bounded by two parallel hyperplanes. When the inequalities associated with $y(k)$ are in the standard form, these hyperplanes are defined by

$$\mathbb{H}^+ = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid y(k) - \mathbf{r}^{\mathrm{T}}(k{-}1)\mathbf{p} = 1\}$$

and

$$\mathbb{H}^- = \{\mathbf{p} \in \mathbb{R}^{n_p} \mid y(k) - \mathbf{r}^T(k-1)\mathbf{p} = -1\}.$$

When $\mathbf{r}(k-1) = \mathbf{0}$, $y(k)$ conveys no information on $\mathbf{p}$. We therefore only consider data points with non-zero regressors. As the intersection with $\mathbb{P}_0$ of strips bounded by parallel hyperplanes, $\mathbb{P}_{n_t}$ is a convex polyhedron. Provided the $\mathbf{r}(k)$'s ($k = 0, \dots, n_t - 1$) span $\mathbb{R}^{n_p}$ (a condition for identifiability), $\mathbb{P}_{n_t}$ is bounded, and thus a polytope, even if $\mathbb{P}_0$ is $\mathbb{R}^{n_p}$.

Conditions under which $\mathbb{P}_{n_t}$ will converge to $\mathbf{p}^*$ as n_t tends to infinity are investigated in (Veres and Norton, 1991a). If $\mathbf{p}^*$ is such that the bounds on the error are satisfied, an important condition for excluding all $\mathbf{p} \neq \mathbf{p}^*$ as n_t increases is that enough error samples are close to the bounds.

$\mathbb{P}_{n_t}$ may become very complicated if n_t and especially n_p are large. This will be particularly true when the error bounds are pessimistic, which is the rule in practice since an optimistic choice of bounds yields an empty set $\mathbb{P}_{n_t}$ after a finite number of observations. It is therefore of special interest to have at one's disposal methods for the construction of sets with simple shapes guaranteed to enclose $\mathbb{P}_{n_t}$. The most widely used are ellipsoids, orthotopes (boxes), parallelotopes and polyhedra with limited complexity (*e.g.* simplexes).

5.4.1.1 Recursive determination of outer ellipsoids

In this approach, the data are taken into account one after the other to construct a succession of ellipsoids containing all values of $\mathbf{p}$ consistent with all previous measurements. This leads to a policy easily implementable on-line (possibly in real time). The algorithms obtained can be regarded as members of the wider family of *dead-zone algorithms* (Arruda and Favier, 1991). After the first $k - 1$ observations, $\mathbb{P}_{k-1}$ is characterized by the ellipsoid

$$\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1}) = \{\mathbf{p} \in \mathbb{R}^{n_p} \mid (\mathbf{p} - \hat{\mathbf{p}}^{k-1})^T\mathbf{M}_{k-1}^{-1}(\mathbf{p} - \hat{\mathbf{p}}^{k-1}) \leq 1\},$$

where $\hat{\mathbf{p}}^{k-1}$ is the centre of the ellipsoid, and $\mathbf{M}_{k-1}$ a positive-definite matrix which specifies its size and orientation. The volume of the uncertainty region for $\mathbf{p}$ characterized by this ellipsoid is given by

$$\text{vol}\ \mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1}) = \mathcal{V}(n_p)\sqrt{\det \mathbf{M}_{k-1}},$$

where $\mathcal{V}(n_p)$ is the volume of the unit ball in $\mathbb{R}^{n_p}$.

The algorithms described below provide rules for computing $\hat{\mathbf{p}}^k$ and $\mathbf{M}_k$ in such a way that

$$\mathbb{E}(\hat{\mathbf{p}}^k, \mathbf{M}_k) \supseteq \mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1}) \cap \mathbb{S}_k,$$

while minimizing the volume of $\mathbb{E}(\hat{\mathbf{p}}^k, \mathbf{M}_k)$, which amounts to minimizing det $\mathbf{M}_k$ (Figure 5.12). The initial values $\hat{\mathbf{p}}^0$ and $\mathbf{M}_0$ are chosen so as to ensure that $\mathbb{E}(\hat{\mathbf{p}}^0, \mathbf{M}_0)$ is large enough to contain $\mathbb{P}_{n_t}$ (*e.g.* $\hat{\mathbf{p}}^0 = \mathbf{0}$ and $\mathbf{M}_0 = c\mathbf{I}_{n_p}$ with c large enough).

We assume here that the parameters do not vary with k, so that $\mathbb{P}_{k-1} \supseteq \mathbb{P}_k$, but the approach extends to the tracking of varying parameters, provided the ellipsoid $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1})$ is suitably expanded before being intersected with $\mathbb{S}_k$ (Norton and Mo, 1990).

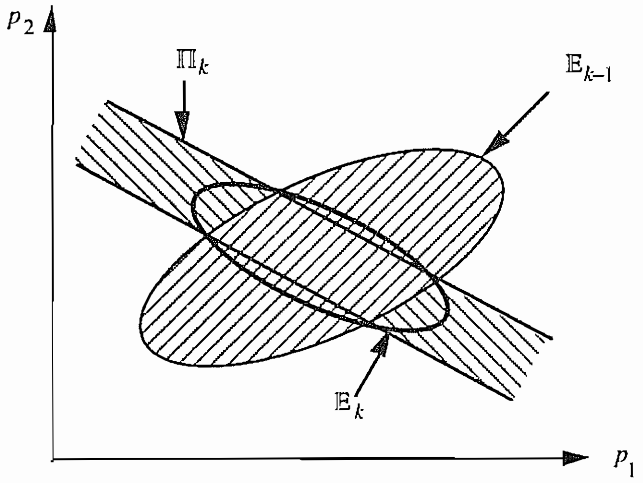


Figure 5.12. Principle of the recursive determination of outer ellipsoids

OBE (Outer-Bounding Ellipsoid) algorithm. This algorithm, originating from the work of Schweppe (1968, 1973), Fogel and Huang (1982) and Belforte, Bona and Cerone (1990), is the most widely used for the recursive ellipsoidal characterization of posterior feasible sets. We shall derive its equations and detail the tests required for its implementation. First note that

$$\mathbf{p} \in \mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1}) \cap \mathbb{\Pi}_k$$

if and only if

$$(\mathbf{p} - \hat{\mathbf{p}}^{k-1})^{\mathrm{T}}\mathbf{M}_{k-1}^{-1}(\mathbf{p} - \hat{\mathbf{p}}^{k-1}) \leq 1 \quad \text{and} \quad [y(k) - \mathbf{r}^{\mathrm{T}}(k-1)\mathbf{p}]^2 \leq 1.$$

Any $\mathbf{p}$ in $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1}) \cap \mathbb{\Pi}_k$ therefore satisfies

$$(\mathbf{p} - \hat{\mathbf{p}}^{k-1})^{\mathrm{T}}\mathbf{M}_{k-1}^{-1}(\mathbf{p} - \hat{\mathbf{p}}^{k-1}) + \lambda[y(k) - \mathbf{r}^{\mathrm{T}}(k-1)\mathbf{p}]^2 \leq 1 + \lambda, \quad \forall\, \lambda \geq 0.$$

For any fixed λ, this is only a *necessary* condition for $\mathbf{p}$ to belong to $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1}) \cap \mathbb{\Pi}_k$. This condition is quadratic in $\mathbf{p}$ and defines a *family* of ellipsoids $\mathbb{E}(\hat{\mathbf{p}}^{k}, \mathbf{M}_{k})$ parametrized by λ. Various policies can be considered for choosing λ, and several ellipsoidal algorithms with different values of λ can even be run in parallel to characterize posterior feasible sets by the intersection of the ellipsoids thus obtained (Kurzhanski and Valyi, 1991). To calculate the value of λ that minimizes the volume of $\mathbb{E}(\hat{\mathbf{p}}^{k}, \mathbf{M}_{k})$, it is convenient to rewrite the equation that defines this ellipsoid in the standard form. Since

$$\mathbf{r}^{\mathrm{T}}(k-1)\mathbf{p} = \mathbf{r}^{\mathrm{T}}(k-1)\hat{\mathbf{p}}^{k-1} + \mathbf{r}^{\mathrm{T}}(k-1)(\mathbf{p} - \hat{\mathbf{p}}^{k-1}),$$

$\mathbb{E}(\hat{\mathbf{p}}^{k}, \mathbf{M}_{k})$ can be defined by

$$(\mathbf{p}-\hat{\mathbf{p}}^{k-1})^{\mathrm{T}}\overline{\mathbf{M}}_{k-1}^{-1}(\mathbf{p}-\hat{\mathbf{p}}^{k-1})-2\lambda[y(k)-\mathbf{r}^{\mathrm{T}}(k-1)\hat{\mathbf{p}}^{k-1}]\mathbf{r}^{\mathrm{T}}(k-1)(\mathbf{p}-\hat{\mathbf{p}}^{k-1})+$$
$$\lambda[y(k)-\mathbf{r}^{\mathrm{T}}(k-1)\hat{\mathbf{p}}^{k-1}]^2\leq 1+\lambda,$$

with

$$\overline{\mathbf{M}}_{k-1}^{-1}=\mathbf{M}_{k-1}^{-1}+\lambda\mathbf{r}(k-1)\mathbf{r}^{\mathrm{T}}(k-1).$$

The matrix-inversion lemma implies:

$$\overline{\mathbf{M}}_{k-1}=[\mathbf{I}_{n_\mathrm{p}}-\frac{\mathbf{M}_{k-1}\mathbf{r}(k-1)\mathbf{r}^{\mathrm{T}}(k-1)}{\lambda^{-1}+\mathbf{r}^{\mathrm{T}}(k-1)\mathbf{M}_{k-1}\mathbf{r}(k-1)}]\,\mathbf{M}_{k-1}.$$

By completing the square in the quadratic form, one obtains:

$$\{\mathbf{p}-\hat{\mathbf{p}}^{k-1}-\lambda\overline{\mathbf{M}}_{k-1}\mathbf{r}(k-1)[y(k)-\mathbf{r}^{\mathrm{T}}(k-1)\hat{\mathbf{p}}^{k-1}]\}^{\mathrm{T}}\overline{\mathbf{M}}_{k-1}^{-1}$$
$$\times\{\mathbf{p}-\hat{\mathbf{p}}^{k-1}-\lambda\overline{\mathbf{M}}_{k-1}\mathbf{r}(k-1)[y(k)-\mathbf{r}^{\mathrm{T}}(k-1)\hat{\mathbf{p}}^{k-1}]\}\leq c_1(\lambda),$$

with

$$c_1(\lambda)=1+\lambda-\lambda[y(k)-\mathbf{r}^{\mathrm{T}}(k-1)\hat{\mathbf{p}}^{k-1})]^2[1-\lambda\mathbf{r}^{\mathrm{T}}(k-1)\overline{\mathbf{M}}_{k-1}\mathbf{r}(k-1)]$$
$$=1+\lambda-\lambda[y(k)-\mathbf{r}^{\mathrm{T}}(k-1)\,\hat{\mathbf{p}}^{k-1}]^2[1+\lambda\mathbf{r}^{\mathrm{T}}(k-1)\mathbf{M}_{k-1}\mathbf{r}(k-1)]^{-1}.$$

Define $v=y(k)-\mathbf{r}^{\mathrm{T}}(k-1)\hat{\mathbf{p}}^{k-1}$ and $g=\mathbf{r}^{\mathrm{T}}(k-1)\mathbf{M}_{k-1}\mathbf{r}(k-1)$ ($g>0$ since the regressor vector is assumed not to be zero). Then

$$c_1(\lambda)=1+\lambda-\frac{\lambda v^2}{1+\lambda g}.$$

Normalizing the quadratic form gives $\mathbb{E}(\hat{\mathbf{p}}^k,\mathbf{M}_k)$ as a function of λ, with

$$\hat{\mathbf{p}}^k(\lambda)=\hat{\mathbf{p}}^{k-1}+\lambda\overline{\mathbf{M}}_{k-1}(\lambda)\mathbf{r}(k-1)v$$

and

$$\mathbf{M}_k(\lambda)=c_1(\lambda)\overline{\mathbf{M}}_{k-1}(\lambda).$$

Minimizing the volume of $\mathbb{E}(\hat{\mathbf{p}}^k,\mathbf{M}_k)$ with respect to λ is equivalent to minimizing

$$\det\mathbf{M}_k(\lambda)=[c_1(\lambda)]^{n_\mathrm{p}}\det\overline{\mathbf{M}}_{k-1}(\lambda).$$

Since $\det[\mathbf{I}+\mathbf{v}\mathbf{w}^{\mathrm{T}}]=1+\mathbf{w}^{\mathrm{T}}\mathbf{v}$,

$$\det\mathbf{M}_k(\lambda)=c_2(\lambda)\det\mathbf{M}_{k-1},$$

where

$$c_2(\lambda) = \frac{[c_1(\lambda)]^{n_\mathrm{p}}}{1 + \lambda \mathbf{r}^\mathrm{T}(k-1)\mathbf{M}_{k-1}\mathbf{r}(k-1)} = \frac{\left(1 + \lambda - \dfrac{\lambda \nu^2}{1 + \lambda g}\right)^{n_\mathrm{p}}}{1 + \lambda g}.$$

The minimum-volume ellipsoid *in the family considered* is thus obtained for

$$\lambda^* = \arg \min_{\lambda \geq 0} c_2(\lambda).$$

For all $n_\mathrm{p} > 1$, $c_2(\lambda)$ tends to $+\infty$ with λ, so λ^* is either zero or the argument of a stationary point of $c_2(\lambda)$. These stationary points are given by

$$\frac{\mathrm{d}}{\mathrm{d}\lambda} c_2(\lambda) = 0,$$

which is equivalent to $c_3(\lambda)c_4(\lambda) = 0$, where

$$c_3(\lambda) = g\lambda^2 + (1 + g - \nu^2)\lambda + 1,$$

$$c_4(\lambda) = (n_\mathrm{p} - 1)g^2\lambda^2 + g(2n_\mathrm{p} - 1 - g + \nu^2)\lambda + n_\mathrm{p}(1 - \nu^2) - g.$$

The equation $c_3(\lambda) = 0$ has no positive real root, since $\nu^2 < 1$ if $\hat{\mathbf{p}}^{k-1} \in \mathbb{S}_k$. The stationary points are thus the real roots of the equation $c_4(\lambda) = 0$, which can also be written as

$$\alpha_1 \lambda^2 + \alpha_2 \lambda + \alpha_3 = 0,$$

where $\alpha_1 = (n_\mathrm{p} - 1)g^2 > 0$. Define

$$a_+ = \frac{\nu - 1}{\sqrt{g}} \quad \text{and} \quad a_- = -\frac{\nu + 1}{\sqrt{g}}.$$

The discriminant of the equation $c_4(\lambda) = 0$ can then be written as

$$\Delta = \frac{g^4}{4}\left[4(1 - a_+^2)(1 - a_-^2) + n_\mathrm{p}^2 (a_+^2 - a_-^2)^2\right].$$

The indicators a_+ and a_- have several interesting properties, shown in Figure 5.13:

— $|a_+| > 1 \Leftrightarrow \mathbb{H}^+$ does not cut $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1})$;
— $|a_-| > 1 \Leftrightarrow \mathbb{H}^-$ does not cut $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1})$;
— $a_+ > 1$ or $a_- > 1 \Leftrightarrow \mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1}) \cap \mathbb{S}_k$ is empty $\Rightarrow \mathbb{P}_k$ is empty; (this test should be used to stop the algorithm if required;)
— $\mathrm{d}c_2(\lambda)/\mathrm{d}\lambda|_{\lambda=0} = g[n_\mathrm{p}a_+a_- - 1]$.

A sufficient condition for λ^* to be a stationary point is

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}c_2(\lambda)\Big|_{\lambda=0} = n_\mathrm{p}(1-\nu^2) - g < 0,$$

which is thus equivalent to $a_+ a_- < 1/n_\mathrm{p}$.

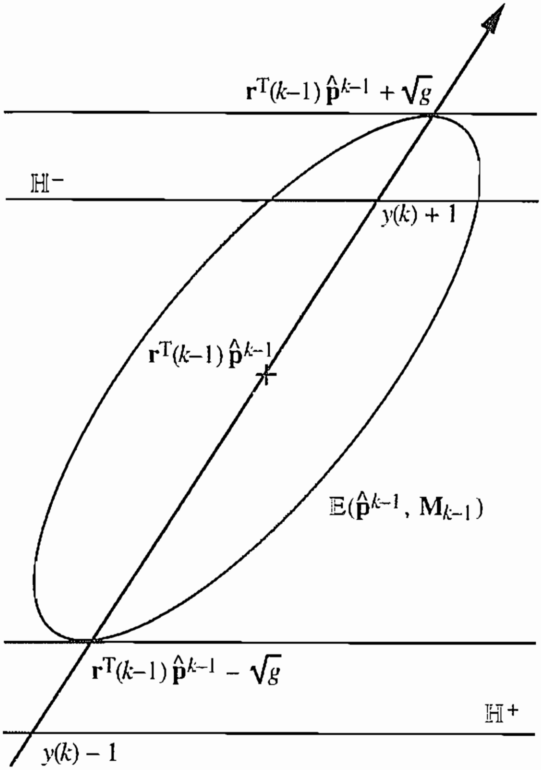

$\mathbf{r}^\mathrm{T}(k-1)\,\hat{\mathbf{p}}^{k-1} + \sqrt{g}$
$\mathbb{H}^-$
$y(k) + 1$
$\mathbf{r}^\mathrm{T}(k-1)\,\hat{\mathbf{p}}^{k-1}$
$\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1})$
$\mathbf{r}^\mathrm{T}(k-1)\,\hat{\mathbf{p}}^{k-1} - \sqrt{g}$
$\mathbb{H}^+$
$y(k) - 1$

Figure 5.13. Illustration of the properties of the indicators a_+ and a_-:
$y(k) - 1 < r^\mathrm{T}(k-1)\hat{\mathrm{p}}^{k-1} - \sqrt{g}$, so $a_+ < -1$;
$r^\mathrm{T}(k-1)\hat{\mathrm{p}}^{k-1} - \sqrt{g} < y(k) + 1 < r^\mathrm{T}(k-1)\hat{\mathrm{p}}^{k-1} + \sqrt{g}$, so $-1 < a_- < 1$

When $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1}) \cap \mathbb{I}_k$ is not empty, any hyperplane not intersecting $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1})$ can obviously be replaced by a parallel one tangent to this ellipsoid without modifying the intersection between the ellipsoid and the feasible strip $\mathbb{I}_k$. Belforte, Bona and Cerone (1990) have, however, found that this policy yields smaller ellipsoids than the original algorithm proposed by Fogel and Huang (1982), which is therefore suboptimal. The OBE algorithm considered below implements this heuristic modification, which will turn out to yield an optimal algorithm. It amounts to replacing a_- by $a'_- = \max\,(a_-, -1)$ and a_+ by $a'_+ = \max\,(a_+, -1)$. Whenever a_+ or a_- have been modified thus, ν must be updated to

$$\nu' = \frac{a'_- - a'_+}{a'_+ + a'_-}\ ,$$

yielding $\nu' = \sqrt{g} - 1$ if a_- has been replaced by $a'_- = -1$ or $\nu' = 1 - \sqrt{g}$ if a_+ has been replaced by $a'_+ = -1$. In what follows, the primes are omitted for notational convenience.

The discriminant Δ is then always strictly positive, which implies that the equation $c_4(\lambda) = 0$ always has two real roots. Moreover,

$$\alpha_2 = g^2\Big[\frac{n_p - 1}{2}(a_+^2 + a_-^2) + (n_p a_+ a_- - 1) + \frac{2\nu^2}{g}\Big],$$

and

$$\alpha_3 = g^2\Big(\frac{a_+ + a_-}{2}\Big)^2 (n_p a_+ a_- - 1).$$

Two cases must be distinguished. If $a_+ \mathrm{a}_- \geq 1/n_p$ then $\alpha_3 \geq 0$ and $\alpha_2 > 0$, which implies that no root is strictly positive and thus that $c_2(\lambda)$ has no stationary point in $[0, +\infty[$, so $\lambda^* = 0$. Conversely, if $a_+ a_- < 1/n_p$, which implies that $\alpha_3 < 0$, then there is a single positive root

$$\lambda^* = \frac{-\alpha_2 + \sqrt{\Delta}}{2\alpha_1}.$$

The new ellipsoid $\mathbb{E}(\hat{\mathbf{p}}^k, \mathbf{M}_k)$ is obtained by substituting λ^* for λ in the expressions giving $\hat{\mathbf{p}}^k$ and $\mathbf{M}_k$. Note that $\lambda^* = 0$ corresponds to retaining the previous ellipsoid.

EPC (Ellipsoid with Parallel Cuts) algorithm. EPC is one of the ellipsoidal algorithms arising from study of the theoretical complexity of linear-programming algorithms (Khachiyan, 1979; Bland, Goldfarb and Todd, 1981). It calculates the minimum-volume ellipsoid containing $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1}) \cap \mathbb{\Pi}_k$, where $\mathbb{\Pi}_k$ is the strip between two parallel hyperplanes $\mathbb{H}^+$ and $\mathbb{H}^-$, and applies when both $\mathbb{H}^+$ and $\mathbb{H}^-$ cut the ellipsoid $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1})$. It may thus be used for bounded-error estimation provided:

— data with a regressor equal to zero are not considered,
— a test is used to check that the intersection is not empty,
— any hyperplane not intersecting $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1})$ is translated to become tangent to it.

As for OBE, this amounts to performing the following preliminary operations:

— if $\mathbf{r}(k-1) = 0$, then $\mathbb{E}(\hat{\mathbf{p}}^k, \mathbf{M}_k) = \mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1})$;
— if $a_- > 1$ or $a_+ > 1$, then $\mathbb{E}(\hat{\mathbf{p}}^k, \mathbf{M}_k)$ is empty; else replace a_- by $\max(a_-, -1)$ and a_+ by $\max(a_+, -1)$;
— if $a_+ a_- \geq 1/n_p$ then $\mathbb{E}(\hat{\mathbf{p}}^k, \mathbf{M}_k) = \mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1})$.

When $\mathbb{E}(\hat{\mathbf{p}}^k, \mathbf{M}_k)$ is not given by these preliminary operations, it is calculated, if $a_+ \neq a_-$, by

$$\hat{\mathbf{p}}^k = \hat{\mathbf{p}}^{k-1} + \frac{\alpha(a_+ - a_-)}{2\sqrt{g}} \mathbf{M}_{k-1}\mathbf{r}(k-1),$$

$$\mathbf{M}_k = \delta\Big[\mathbf{M}_{k-1} - \frac{\alpha}{g}\mathbf{M}_{k-1}\mathbf{r}(k-1)\mathbf{r}^{\mathrm{T}}(k-1)\mathbf{M}_{k-1}\Big]$$

where

$$\delta = \frac{n_p^2}{n_p^2 - 1}\Big[1 - \frac{a_+^2 + a_-^2 - \frac{\rho}{n_p}}{2}\Big]$$

and

$$\alpha = \frac{1}{n_p + 1}\Big[n_p + \frac{2}{(a_+ - a_-)^2}\big(1 - a_+a_- - \frac{\rho}{2}\big)\Big],$$

with

$$\rho = \sqrt{4(1 - a_+^2)(1 - a_-^2) + n_p^2(a_+^2 - a_-^2)^2}.$$

When $a_+ = a_- = a$, α is no longer defined. The equations above then specialize into those of the *ellipsoidal algorithm with parallel cuts symmetric with respect to the centre*, given by

$$\hat{\mathbf{p}}^k = \hat{\mathbf{p}}^{k-1}$$

and

$$\mathbf{M}_k = \frac{n_p(1 - a^2)}{n_p - 1}\Big[\mathbf{M}_{k-1} - \frac{1 - n_p a^2}{(1 - a^2)g}\mathbf{M}_{k-1}\mathbf{r}(k-1)\mathbf{r}^{\mathrm{T}}(k-1)\mathbf{M}_{k-1}\Big].$$

EPC yields the minimum-volume ellipsoid containing $\mathbb{E}(\hat{\mathbf{p}}^{k-1}, \mathbf{M}_{k-1}) \cap \mathbb{H}_k$ (König and Pallaschke, 1981). It is thus *recursively* optimal. Now, although the equations of EPC and OBE look quite different, the two algorithms are mathematically equivalent (Pronzato, Walter and Piet-Lahanier, 1989), which establishes that OBE is also *recursively* optimal. OBE (and thus EPC) is surprisingly similar to the recursive least-squares algorithm (RLS), considered in Section 4.1.4. However, OBE and EPC are not simple variants of RLS. As Schweppe (1973) puts it: "The models are fundamentally different both in terms of physical assumptions and interpretations and in terms of type of mathematical concepts required. However, when ellipsoidal sets are used, the unknown-but-bounded and stochastic model equations are very similar in appearance".

REMARKS 5.6

— The convergence properties of $\mathbb{E}_k$ are studied in (Liu, Nayeri and Deller, 1994; Nayeri, Liu and Deller, 1994).

— The fact that OBE and EPC are mathematically equivalent does not imply that they are computationally so. No result seems available yet on their relative numerical efficiency (speed, robustness...).

— An optimal-volume-ellipsoid algorithm is suggested in (Cheung, Yurkovitch and Passino, 1993), presented in such a way that it might seem to differ from OBE and EPC. However, one can easily show (Pronzato and Walter, 1996b) that these three algorithms are mathematically equivalent and only differ in the way the computation is organized.

— Except in the degenerate case, the intersection of $\mathbb{E}_{k-1}$ with $\mathbb{H}_k$ is not an ellipsoid. An approximation is thus incurred at each step, even with recursively optimal algorithms. The ellipsoid $\mathbb{E}(\hat{\mathbf{p}}^{n_t}, \mathbf{M}_{n_t})$ obtained by processing all the data is therefore *not* the minimum-volume ellipsoid containing $\mathbb{P}_{n_t}$. The volume of the ellipsoid can generally be decreased by circulating the data several times in the algorithm (Belforte, Bona and Cerone, 1990). Even in this case, the resulting ellipsoid remains very pessimistic, and the parametric uncertainty intervals (PUI's) obtained by projecting the ellipsoid on the axes of the parametric space are very pessimistic too. Note that better PUI's can be obtained by using the most interior bounds given by such projections during the course of all recursion steps so far (Obali, 1993).

— Techniques inspired by experiment design (Chapter 6) permit computation of the minimum-volume ellipsoid containing $\mathbb{P}_{n_t}$ (Pronzato and Walter, 1994a, 1994b). The resulting algorithms are then globally optimal, see Example 5.4 below, but no longer recursive. An intermediate approach consists in replacing Π_k by the minimum-volume ellipsoid containing the parallelotope associated with the last n_p data (Veres and Norton, 1991b).

— The algorithm presented in (Norton, 1989) recursively determines a maximum-volume ellipsoid contained in $\mathbb{P}_k$. However, this algorithm is not globally optimal, and the ellipsoid obtained tends to vanish quickly. Non-recursive globally optimal algorithms (in the sense of maximal volume) are presented, *e.g.*, in (Khachiyan and Todd, 1993; Pronzato and Walter, 1993, 1996a). See also Example 5.4 below.

— Minimizing the volume of an ellipsoid amounts to minimizing the product of the lengths of its axes, which may lead to a very thin ellipsoid and to large uncertainty in each parameter. One might then prefer to minimize the trace of $\mathbf{M}_k$, which amounts to minimizing the sum of the squares of the lengths of the axes (Fogel and Huang, 1982), or consider more general costs, using the eigenvalues of $\mathbf{M}_k$ (Kiselev and Polyak, 1991).

— In the same way as the recursive least-squares algorithm forms a basic ingredient of Kalman filtering, ellipsoidal outer bounding can be used in bounded-error state estimation (Schweppe, 1968, 1973; Filippova *et al.*, 1996; Maksarov and Norton, 1996a, 1996b; Durieu, Polyak and Walter, 1996a, 1996b). ◊

EXAMPLE 5.4

Consider the AR system defined by

$$y(k) = -0.4y(k-1) - 0.85y(k-2) + e(k),\ k = 3, \dots, 25,$$

$$y(1) = e(1), \quad y(2) = e(2),$$

with the $e(k)$'s independently uniformly distributed in $[-1, 1]$. Figure 5.14 presents the outer ellipsoids obtained for the AR model

$$y(k) = p_1 y(k-1) + p_2 y(k-2) + e(k),$$

$$-1 \leq e(k) \leq 1,\ k = 3, \dots, 25,$$

with the EPC algorithm after 1 to 10 circulations of the data. The true value $\mathbf{p}^* = (-0.4, -0.85)^{\mathrm{T}}$ of the parameters of the model is indicated by a star, and the exact polytope $\mathbb{P}_{25}$ is in solid lines. The ellipsoid obtained after 100 circulations of the data in the EPC algorithm is shown in Figure 5.15, together with the minimum-volume outer ellipsoid enclosing $\mathbb{P}_{25}$, which illustrates the suboptimality of the former. Finally, Figure 5.16 presents the minimum-volume outer ellipsoid and maximum-volume inner ellipsoid for $\mathbb{P}_{25}$. Both are obtained using techniques inspired by the methodology of experiment design (Pronzato and Walter, 1996a). ◊

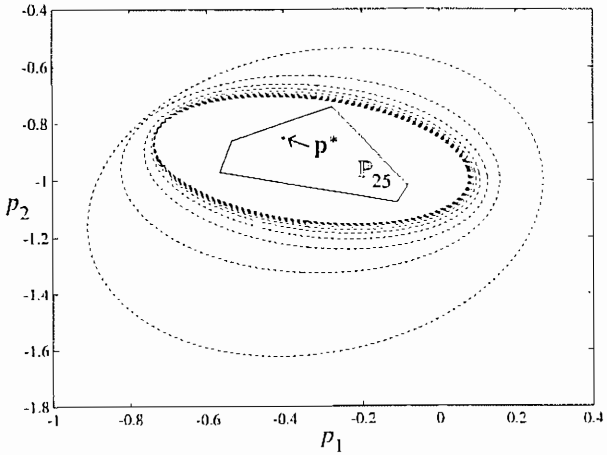


Figure 5.14. Outer ellipsoids obtained in Example 5.4 by circulating the data 1 to 10 times in the EPC algorithm; the exact feasible set $\mathbb{P}_{25}$ is in solid lines

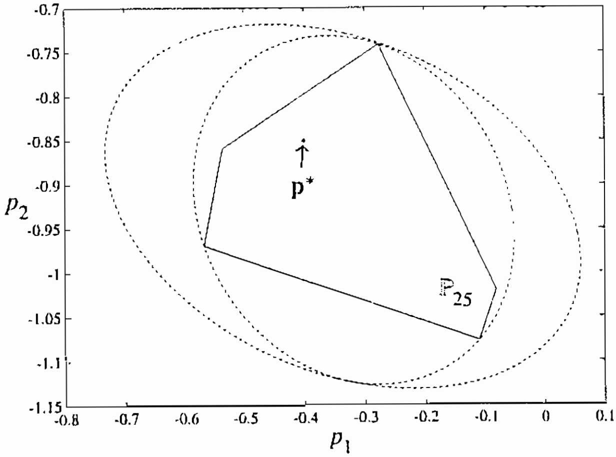


Figure 5.15. Minimum-volume outer ellipsoid and outer ellipsoid obtained in Example 5.4 by circulating the data 100 times in the EPC algorithm; the exact feasible set $\mathbb{P}_{25}$ is in solid lines

In summary. OBE and EPC

— are mathematically equivalent,
— are recursive and easy to implement in real time,
— are recursively (but not globally) optimal,
— easily extend to the tracking of time-varying parameters,
— require simple calculations (comparable to RLS),

— yield a pessimistic characterization of $\mathbb{P}_{n_t}$,
— yield a pessimistic characterization of the PUI's,
— do not always detect that $\mathbb{P}_{n_t}$ is empty (because they are globally suboptimal).

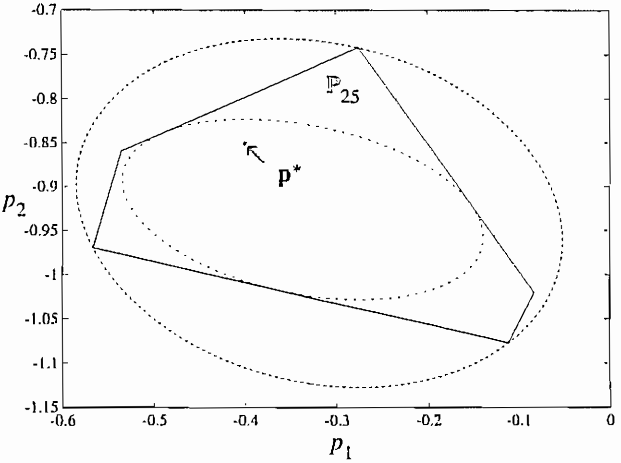


Figure 5.16. Minimum-volume outer ellipsoid and maximum-volume inner ellipsoid for Example 5.4; the exact feasible set $\mathbb{P}_{25}$ is in solid lines

5.4.1.2 Non-recursive determination of outer boxes

Determination of the PUI's of $\mathbf{p}$ amounts to finding the smallest (volume) box with edges parallel to the axes enclosing $\mathbb{P}_{n_t}$. The complexity of the resulting description of $\mathbb{P}_{n_t}$ is moderate: $2n_p$ scalars for a box, versus $n_p(n_p + 3)/2$ for an ellipsoid. To turn the problem of computing each PUI into an optimization, notice that the bounds of the ith PUI are given by the minimal and maximal values of the cost $j(\mathbf{p}) = p_i$, when the feasible domain for $\mathbf{p}$ is $\mathbb{P}_{n_t}$, which is defined by linear inequalities. The cost and constraints being linear, the extremal values can be computed by linear programming. Determination of the box thus requires solution of $2n_p$ linear-programming problems (Milanese and Belforte, 1982), with $2n_t$ linear constraints each (in addition to the constraints defining $\mathbb{P}_0$, if any). Dantzig's (1963) simplex algorithm may be used, as well as more recent techniques; see (Gonzaga, 1992) and Section 4.3.4.1.

In summary. The resulting policy

— is not recursive,
— is ill suited to the tracking of time-varying parameters,
— requires far heavier computation than OBE or EPC,
— yields exact PUI's,
— yields a pessimistic characterization of $\mathbb{P}_{n_t}$,
— detects if $\mathbb{P}_{n_t}$ is empty.

REMARKS 5.7

— Usually, only a few constraints are active on the boundary of $\mathbb{P}_{n_t}$. Preprocessing of the data by OBE or EPC then permits elimination of many redundant constraints, thereby considerably simplifying computation.

— When $\mathbb{P}_{n_t}$ is thin and badly oriented, using a box with edges parallel to the axes of the parametric space yields a very poor characterization of $\mathbb{P}_{n_t}$. Replacing these axes by those of an outer ellipsoid, determined by OBE or EPC, will much improve the situation.

— Outer boxes can also be determined recursively by calculating, at each recursion step k, the minimum-volume box with edges parallel to the axes containing the intersection of $\mathbb{H}_k$ with the previous box (Pshenichnyy and Pokotilo, 1983; Messaoud, Favier and Santos Mendes, 1992). This solution can easily be adapted to parameter tracking. Parallelotopes can also be determined recursively (Vicino and Zappa, 1992), as can polytopes with limited complexity (Piet-Lahanier and Walter, 1993). As already noted for ellipsoids, recursively optimal algorithms are generally not globally optimal. ◊

5.4.1.3 Exact description

Assume that $\mathbb{P}_0$ is either $\mathbb{R}^{n_p}$ or defined by a finite number of linear inequalities. When the error is affine in $\mathbf{p}$, $\mathbb{P}_{n_t}$ can then be written as

$$\mathbb{P}_{n_t} = \{\mathbf{p} \mid \mathbf{Ap} \geq \mathbf{b}\}.$$

Let $\mathbf{a}_i^{\mathrm{T}}$ denote the ith row of $\mathbf{A}$. When not empty, $\mathbb{P}_{n_t}$ is a convex polyhedron, which can be computed recursively (Walter and Piet-Lahanier, 1989; Broman and Shensa, 1990; Mo and Norton, 1990; Kuntzevich and Lychak, 1992; Piet-Lahanier and Walter, 1994). The inequalities are taken into account one after the other, each observation providing two inequalities. We restrict ourselves to basic ideas of the algorithm in the case where the polyhedron is bounded. The general situation is treated in (Walter and Piet-Lahanier, 1989).

Assume that the polyhedron $\mathbb{Q}_{k-1}$ formed by the first $(k-1)$ inequalities is as in Figure 5.17. The non-trivial case is when only a part of the previous polyhedron is consistent with the kth inequality, as in Figure 5.18. Some vertices should then be kept (here 1, 4 and 5), whereas others should be removed (here 2 and 3). Moreover, a new vertex should be created on each edge connecting a retained vertex and an *adjacent* removed vertex (giving 2' and 3' here). (No vertex should be created between 5 and 3 as they are not adjacent.)

A representation of the polyhedron facilitating determination of adjacent vertices should thus be updated. $\mathbb{Q}_k$ will be characterized as the convex hull of its vertices $\mathbf{v}_i$ $(i = 1, \ldots, I_k)$. With $\mathbb{Q}_k$ we associate a matrix $\mathbf{S}_k$, the columns of which contain the vertex coordinates of $\mathbb{Q}_k$. With the ith vertex $\mathbf{v}_i$ are associated lists LAV$_i$ of its adjacent vertices and LSH$_i$ of its supporting hyperplanes. Each vertex is the intersection of at least n_p supporting hyperplanes.

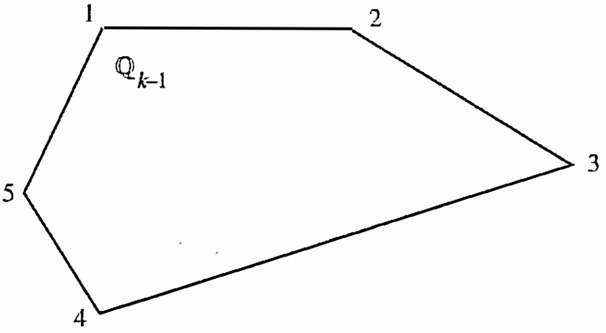


Figure 5.17. Polyhedron formed by the first $(k-1)$ inequalities

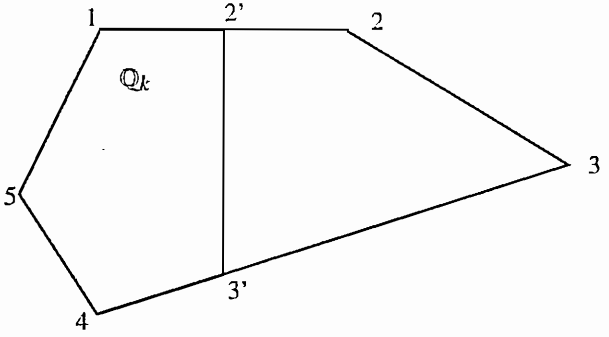


Figure 5.18. Polyhedron formed by the first k inequalities

Initialization. $\mathbb{Q}_0$ may be chosen as a box with edges parallel to the axes, large enough to be sure to contain $\mathbb{Q}_{n_t}$. This simplifies construction of the initial lists of adjacent vertices and supporting hyperplanes.

Iteration. Assume that $\mathbb{Q}_k$ has been determined. The $(k+1)$th inequality defines a feasible half-space

$$\mathbb{H}_{k+1}^{+} = \{\mathbf{p} \mid \mathbf{a}_{k+1}^{\mathrm{T}}\mathbf{p} - b_{k+1} \geq 0\},$$

bounded by the hyperplane $\mathbb{H}_{k+1} = \{\mathbf{p} \mid \mathbf{a}_{k+1}^{\mathrm{T}}\mathbf{p} - b_{k+1} = 0\}$. The updated polyhedron $\mathbb{Q}_{k+1} = \mathbb{Q}_k \cap \mathbb{H}_{k+1}^{+}$ is computed as follows. If no vertex of $\mathbb{Q}_k$ is consistent with the new inequality, $\mathbb{Q}_{k+1}$ is empty and no parameter value is feasible *a posteriori*. If all vertices of $\mathbb{Q}_k$ are consistent with the new inequality, this inequality is redundant, $\mathbb{Q}_{k+1} = \mathbb{Q}_k$, $\mathbf{S}_{k+1} = \mathbf{S}_k$ and the lists are unchanged. Consider now the case where only some vertices are kept.

Updating the vertex matrix. Copy into $\mathbf{S}_{k+1}$ all vertices $\mathbf{v}_i$ of $\mathbb{Q}_k$ that must be kept, *i.e.* with $\mathbf{a}_{k+1}^{\mathrm{T}}\mathbf{v}_i - b_{k+1} \geq 0$. For each, using its LAV$_i$ determine all vertices $\mathbf{v}_n$ of $\mathbb{Q}_k$ adjacent to $\mathbf{v}_i$ and such that

$$\mathbf{a}_{k+1}^{\mathrm{T}}\mathbf{v}_n - b_{k+1} < 0.$$

Create a new vertex $\mathbf{v}_{i,n}$ at the intersection of edge $(\mathbf{v}_i, \mathbf{v}_n)$ with $\mathbb{H}_{k+1}$, given by

$$\mathbf{v}_{i,n} = (1 - \lambda)\mathbf{v}_i + \lambda\mathbf{v}_n, \text{ with } \lambda = \frac{\mathbf{a}_{k+1}^{\mathrm{T}}\mathbf{v}_i - b_{k+1}}{\mathbf{a}_{k+1}^{\mathrm{T}}(\mathbf{v}_i - \mathbf{v}_n)}.$$

Include $\mathbf{v}_{i,n}$ in $\mathbb{S}_{k+1}$.

Updating the LSH's. The LSH's of the vertices of $\mathbb{Q}_k$ kept in $\mathbb{Q}_{k+1}$ are not modified. The LSH of any new vertex $\mathbf{v}_{i,n}$ contains $\mathbb{H}_{k+1}$ and the hyperplanes common to LSH$_i$ and LSH$_n$.

Updating the LAV's. In the LAV of any vertex $\mathbf{v}_i$ kept in $\mathbb{Q}_{k+1}$, replace any vertex $\mathbf{v}_n$ removed from $\mathbb{Q}_k$ by the corresponding new vertex $\mathbf{v}_{i,n}$. Create the LAV associated with each new vertex $\mathbf{v}_{i,n}$, by including the retained vertex $\mathbf{v}_i$ from which it was created, and all new vertices with which $\mathbf{v}_{i,n}$ is adjacent. Each of these new vertices has at least $n_{\mathrm{p}} - 1$ supporting hyperplanes in common with $\mathbf{v}_{i,n}$, and no other vertex has an LSH containing the same $n_{\mathrm{p}} - 1$ hyperplanes.

The exact description is often much simpler than one might fear, because many inequalities are redundant. The method can be extended to tracking time-varying parameters, provided the previous polyhedron is expanded at each iteration, to account for possible evolution of parameters, before intersecting it with the new feasible half-space. Piet-Lahanier and Walter (1993, 1994) suggest an expansion that does not modify the lists LAV and LSH, which greatly simplifies the implementation. The complexity of the description obtained at each iteration can also be restricted. Finally, the extension of this method to polyhedral cones permits recursive determination of a minimax estimator for **p** (*i.e.* the smallest bound on the errors such that $\mathbb{P}_{n_t}$ is not empty and the associated set $\mathbb{P}_{n_t}$, in general a singleton), together with all sets $\mathbb{P}_{n_t}$ associated with larger error bounds (Walter and Piet-Lahanier, 1991). This proves particularly useful when there is no prior information on the value of the bound (Example 4.16).

REMARK 5.8

A situation where $\mathbb{P}_k$ is empty ($k \leq n_t$),

— will be detected with no delay by the exact description;
— will be detected once all data have been processed by the non-recursive outer-box approach;
— may go undetected by the recursive outer-box and outer-ellipsoid approaches. ◊

5.4.2 Non-LP model structures

This situation is the rule for phenomenological models. The posterior feasible set $\mathbb{P}_{n_t}$ is no longer a polyhedron. It may be non-connected, even if the model structure is globally identifiable; see, *e.g.*, (Pronzato and Walter, 1990). The phenomenon can be better understood in the space of observations. Depending on the relative positions of the expectation surface $\mathbb{S}_{\text{exp}}$ (the surface of possible model responses) and the box $\mathbb{Y}$ of responses admissible *a priori*, $\mathbb{P}_{n_t}$ may be connected or not. Consider, for instance a one-parameter model structure with two observations, so that the expectation surface $\mathbb{S}_{\text{exp}}$ is a curve. Assume that $\mathbb{S}_{\text{exp}}$ is as presented in Figure 5.19 and continuous in the parameter p, with each point of $\mathbb{S}_{\text{exp}}$ corresponding to a single value of p (identifiability condition). If the observations are associated with the box $\mathbb{Y}_1$, the posterior feasible set $\mathbb{P}_2$ for p will be connected, whereas it will not be the case if the observations are associated with the box $\mathbb{Y}_2$. This possibility for $\mathbb{P}_2$ not to be connected must be taken into account by the algorithm used for its characterization.

REMARK 5.9

Non-LP structures raise the same type of difficulty when a statistical approach is used to characterize confidence regions for the parameters (Section 5.1.3), or when a point estimate has to be determined (Section 4.3.9.1). ◊

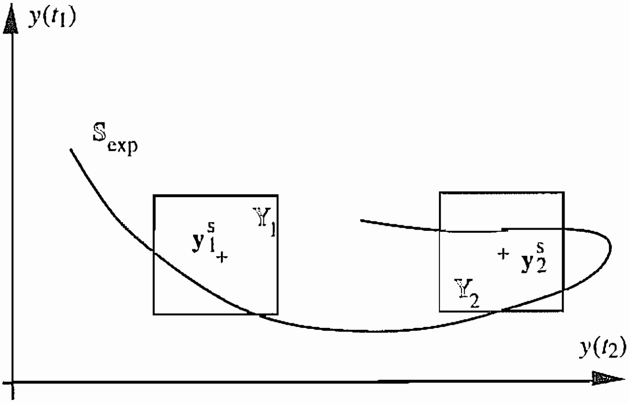


Figure 5.19. Depending on the relative positions of $\mathbb{Y}$ and $\mathbb{S}_{\text{exp}}$, $\mathbb{P}_2$ may be connected or not

EXAMPLE 5.5

Consider the model

$$y_{\text{m}}(t, \mathbf{p}) = p_1 \exp(-p_2 t),$$

with $y(t) = y_{\text{m}}(t, \mathbf{p}^*) + \varepsilon(t)$, $\mathbf{p}^* = (10, 1)^{\text{T}}$, $\mathbb{P}_0 = \mathbb{R}^2$ and the $\varepsilon(t)$'s independently uniformly distributed in $[-0.75, 0.75]$. Figure 5.20 shows the set $\mathbb{P}_{71}$ obtained from the data collected at times $t_i = 0.45 + 0.05i$ ($i = 1, \dots, 71$), when the errors are assumed to lie in $[-1, 1]$. Only six of the 142 inequalities contribute to the definition of $\mathbb{P}_{71}$, and thus correspond to actually useful data, but one cannot know beforehand which data will turn out to be useful. The minimum-volume outer ellipsoid for $\mathbb{P}_{71}$, obtained by techniques inspired by experiment design, is also shown; see (Pronzato and Walter, 1996a) for details. ◊

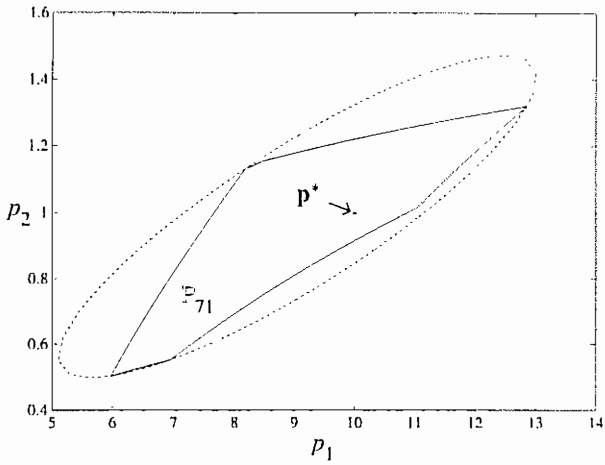


Figure 5.20. Posterior feasible set and minimum-volume ellipsoid for Example 5.5 (non-LP model)

5.4.2.1 Errors in variables

This approach, already considered in a statistical context in Section 4.1.7, allows extension of the techniques developed for LP structures with deterministic regressors to uncertain regressors and/or non-LP structures (Norton, 1987; Clément and Gentil, 1988, 1990; Merkuryev, 1989; Cerone, 1991; Veres and Norton, 1991c). It applies, for instance, when $\mathbf{r}(k)$ contains noisy measurements of the process inputs and outputs.

Define the *regressor error* as

$$\mathbf{e}_{\mathrm{r}}(k, \mathbf{p}) = \mathbf{r}(k) - \mathbf{r}_{\mathrm{m}}(k, \mathbf{p}),$$

where $\mathbf{r}(k)$ is known and $\mathbf{r}_{\mathrm{m}}(k, \mathbf{p})$ is such that $y_{\mathrm{m}}(k+1, \mathbf{p}) = \mathbf{r}_{\mathrm{m}}^{\mathrm{T}}(k, \mathbf{p})\mathbf{p}$. The (pseudo) regressor vector $\mathbf{r}_{\mathrm{m}}$ may depend on $\mathbf{p}$ (non-LP structure), or merely consist of n_{p} unspecified scalars. Assume that the ith component of the regressor error should satisfy

$$e^{\mathrm{m}}_{\mathrm{r}_i}(k) \leq e_{\mathrm{r}_i}(k, \mathbf{p}) \leq e^{\mathrm{M}}_{\mathrm{r}_i}(k), \quad i = 1, \ldots, n_{\mathrm{p}}, k = 0, \ldots, n_{\mathrm{t}} - 1,$$

where the bounds $\mathbf{e}^{\mathrm{m}}_{\mathrm{r}}(k)$ and $\mathbf{e}^{\mathrm{M}}_{\mathrm{r}}(k)$ are known *a priori*. $\mathbb{P}_{n_{\mathrm{t}}}$ then contains all values of $\mathbf{p}$ such that

$$e^{\mathrm{m}}_{\mathrm{y}}(k) - \sum_{i=1}^{n_{\mathrm{p}}} e_{\mathrm{r}_i}(k-1, \mathbf{p})p_i \leq y(k) - \mathbf{r}^{\mathrm{T}}(k-1)\mathbf{p}$$

$$\leq e^{\mathrm{M}}_{\mathrm{y}}(k) - \sum_{i=1}^{n_{\mathrm{p}}} e_{\mathrm{r}_i}(k-1, \mathbf{p})p_i, \quad k = 1, \ldots, n_{\mathrm{t}}.$$

A *necessary* condition for $\mathbf{p}$ to belong to $\mathbb{P}_{n_t}$ is then

$$e_y^{m}(k) - \sum_{i=1}^{n_p} e_{r_i}^{B}(k-1)p_i \leq y(k) - \mathbf{r}^T(k-1)\mathbf{p}$$
$$\leq e_y^{M}(k) - \sum_{i=1}^{n_p} e_{r_i}^{A}(k-1)p_i, \quad k = 1, \ldots, n_t,$$

where

$$[e_{r_i}^{A}(k), e_{r_i}^{B}(k)] = \begin{cases} [e_{r_i}^{m}(k), e_{r_i}^{M}(k)] \text{ if } p_i \geq 0, \\ [e_{r_i}^{M}(k), e_{r_i}^{m}(k)] \text{ if } p_i < 0. \end{cases}$$

For any *given* combination of the signs of the components of $\mathbf{p}$, this corresponds to n_t pairs of linear inequalities. A set containing $\mathbb{P}_{n_t}$ is thus obtained by taking the union of the sets defined by these inequalities for all orthants of parameter space. The bounding hyperplanes for a pair of inequalities in a given orthant are no longer parallel, so OBE and EPC do not apply directly. Specific algorithms, involving a single cut instead of two parallel cuts, may be used (Bland, Golfarb and Todd, 1981; Pronzato and Walter, 1994a), as well as a heuristic modification of OBE based on creating fictitious hyperplanes tangent to the ellipsoid (Clément and Gentil, 1988, 1990). Algorithms computing boxes or polytopes (possibly with restricted complexity) may also be employed.

REMARK 5.10

When the successive errors in the components of the regressor are mutually independent and independent of the output error (which does not hold for models with noisy autoregressive parts), this condition is also *sufficient* (Cerone, 1991, 1993). It is then possible to determine $\mathbb{P}_{n_t}$ exactly by considering linear inequalities. ◊

EXAMPLE 5.6

Consider again the model of Example 4.3

$$y_m(t+1, \mathbf{p}) = -a_1 y_m(t, \mathbf{p}) - a_2 y_m(t-1, \mathbf{p}) - \ldots - a_{n_a} y_m(t+1-n_a, \mathbf{p})$$
$$+ b_1 u_a(t) + \ldots + b_{n_b} u_a(t+1-n_b), \quad t = n_a, \ldots, n_t + n_a - 1,$$

where the inputs $u_a(t)$ actually applied are now assumed to be known only approximately. The input errors satisfy

$$|u(t) - u_a(t)| \leq e_u, \quad t = 1, \ldots, n_t + n_a - 1,$$

where $u(t)$ is the known target value of the input at time t. Similarly, the output errors must satisfy

$$|y(t) - y_m(t, \mathbf{p})| \leq e_y, \quad t = 1, \ldots, n_t + n_a - 1.$$

The bounds e_u and e_y are assumed to be known. This structure with input-output errors is not LP. It may be approximated by an LP structure in the form

$$y_{\mathrm{mlp}}(t+1, \mathbf{p}) = \mathbf{r}^{\mathrm{T}}(t)\mathbf{p},$$

where the parameters are ordered as

$$\mathbf{p} = (a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b})^{\mathrm{T}},$$

and where the unknown regressor vector

$$\mathbf{r}_{\mathrm{m}}(t, \mathbf{p}) = [-y_{\mathrm{m}}(t, \mathbf{p}), \dots, -y_{\mathrm{m}}(t+1-n_a, \mathbf{p}), u_a(t), \dots, u_a(t+1-n_b)]^{\mathrm{T}}$$

is approximated by the known vector

$$\mathbf{r}(t) = [-y(t), \dots, -y(t+1-n_a), u(t), \dots, u(t+1-n_b)]^{\mathrm{T}}.$$

The ith component of the regressor error

$$\mathbf{e}_{\mathrm{r}}(t, \mathbf{p}) = \mathbf{r}(t) - \mathbf{r}_{\mathrm{m}}(t, \mathbf{p}),$$

is then bounded by

$$|e_{\mathrm{r}_i}(t, \mathbf{p})| \leq e^{\mathrm{M}}_{\mathrm{r}_i}(t), \quad i = 1, \dots, n_a + n_b,$$

where

$$e^{\mathrm{M}}_{\mathrm{r}_i}(t) = \begin{cases} e_y & (i = 1, \dots, n_a) \\ e_u & (i = n_a + 1, \dots, n_a + n_b). \end{cases}$$

The set $\mathbb{P}_{n_t}$ defined by the inequalities that can be constructed from the data $y(1), \dots, y(n_t+n_a-1)$ is thus in the set of solutions of

$$-e_y - \sum_{i=1}^{n_a} [e_y \operatorname{sign}(p_i)]p_i - \sum_{k=n_a+1}^{n_a+n_b} [e_u \operatorname{sign}(p_k)]p_k \leq y(t+1) - \mathbf{r}^{\mathrm{T}}(t)\mathbf{p}$$
$$\leq e_y + \sum_{i=1}^{n_a} [e_y \operatorname{sign}(p_i)]p_i + \sum_{k=n_a+1}^{n_a+n_b} [e_u \operatorname{sign}(p_k)]p_k, \quad t = n_a, \dots, n_t + n_a - 1.$$

In a given orthant in parameter space, every $\operatorname{sign}(p_i)$ is constant and this is a set of linear inequalities. ◊

5.4.2.2 Outlier minimal number estimator

Let parameter value $\mathbf{p}$ give a model response satisfying percentage $j(\mathbf{p})$ of the n_t pairs of inequalities. The posterior feasible set is then

$$\mathbb{P}_{n_t} = \{\mathbf{p} \in \mathbb{P}_0 \mid j(\mathbf{p}) = 100\%\}.$$

Maximizing j amounts to minimizing the percentage of inequalities not satisfied, *i.e.* considered as due to outliers, hence the name Outlier Minimal Number Estimator (OMNE) (Lahanier, Walter and Gomeni, 1987; Walter and Piet-Lahanier, 1988).

Basic algorithm

Step 1: Find $\hat{\mathbf{p}} \in \arg\max j(\mathbf{p})$ over $\mathbb{P}_0$.
If $j(\hat{\mathbf{p}}) < 100\%$, then $\mathbb{P}_{n_t}$ is empty (wrong hypotheses), else $\hat{\mathbf{p}} \in \mathbb{P}_{n_t}$.
Step 2: Characterize the boundary of $\mathbb{P}_{n_t}$ by a cloud of points $\{\mathbf{p}_b\}$.

The cost function j is not continuous, and its gradient is zero wherever it is defined; j can therefore not be optimized by any of the local methods presented in Section 4.3.3.

At Step 1, one proceeds to a global optimization over a prior box $\mathbb{P}_0$, for instance using the adaptive-random-search algorithm described in Section 4.3.9.2. Step 2 is based on the techniques presented in Section 5.1.2 for exploring cost contours. Note that $\mathbb{P}_{n_t}$ is the cost contour at level 100%. One is looking for its boundary, which is a hypersurface in parameter space.

REMARK 5.11

By projecting $\{\mathbf{p}_b\}$ onto the axes of parameter space, one obtains an inner estimate of each PUI, that is a lower bound on the uncertainty, as when the Cramér-Rao inequality is used in a statistical context. Note, however, that the Cramér-Rao lower bound is generally valid only when the number of data tends to infinity, whereas the PUI's obtained with OMNE are valid whatever the number of data. Their precision depends, of course, on the ability of the algorithm used to construct the cloud $\{\mathbf{p}_b\}$ to generate points $\mathbf{p}$ with extreme component values. ◊

Consequences of errors in the bounds. When the bounds are pessimistic, $\mathbb{P}_{n_t}$ is too large, but still contains $\mathbf{p}^*$ if the data have been generated by a model with parameters $\mathbf{p}^*$. Conversely, optimistic bounds create *outliers*, *i.e.* data points such that

$$y(k) - y_m(k, \mathbf{p}^*) \notin [e_y^m(k), e_y^M(k)].$$

The presence of outliers will be detected if $j(\mathbf{p}) < 100\%$ for all $\mathbf{p}$'s in $\mathbb{P}_0$. One can then either choose less optimistic bounds (for instance following a minimax estimation procedure), or make the estimator robust to outliers by reducing the number of data the estimate is required to be consistent with. For this purpose, the *feasible set at level* j_0*%* is defined as

$$\mathbb{P}^{j_0} = \{\mathbf{p} \in \mathbb{P}_0 \mid j(\mathbf{p}) \geq j_0\}.$$

OMNE can be used to estimate $\mathbb{P}^{j_0}$, with $j_0 \leq j(\hat{\mathbf{p}})$. Some outliers may not be detected, so $\mathbf{p}^*$ may not belong to the feasible set at level $j(\hat{\mathbf{p}})$%. Protection against n_0 undetected outliers can be achieved, at the possible cost of an increase in the size of the feasible set, by choosing $j_0 = [j(\hat{\mathbf{p}}) - 100 n_0/n_t]\%$. The performance of OMNE in situations with underestimated errors is described in (Walter and Piet-Lahanier, 1988).

The set-inversion technique presented in Section 5.4.2.3 can be extended to characterize $\mathbb{P}^{j_0}$ in a guaranteed way (Jaulin, Walter and Didrit, 1996).

Influence of far outliers. Recall that the breakdown point of an estimator $\hat{\mathbf{p}}$ (Section 3.7.2) is the smallest value of α such that

$$\max_{\mathbf{y}^{\text{o}}} \| \hat{\mathbf{p}}(\mathbf{y}^{\text{s}}) - \hat{\mathbf{p}}(\mathbf{y}^{\text{o}}) \| = \infty,$$

where $\mathbf{y}^{\text{s}}$ is a set of valid data, and $\mathbf{y}^{\text{o}}$ is obtained from $\mathbf{y}^{\text{s}}$ by replacing α% of the data by outliers. When the model is LP and the number of data points tends to infinity, it can be proved (Pronzato and Walter, 1991a, 1996c) that the breakdown point of OMNE tends to 50%, which is the best achievable performance. This estimator has been used for the detection of significant changes between images (Herbin *et al.*, 1989); see Section 3.7.4.

REMARK 5.12

OMNE may behave satisfactorily even in the presence of a large majority of outliers, provided they cannot be described by a model with the structure used. (Note that the least-median-of-squares and least-trimmed-squares estimators do not possess this advantage.) It is only when the outliers are chosen so as to fool the estimator that the 50% upper bound applies. ◊

What to do when the posterior feasible set is not connected? If the problem is due to identifiability, it can be tackled as in Section 5.1.3 by a theoretical study of the structural properties of the model.

EXAMPLE 5.7

We have seen in Example 2.3 that the model structure

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \begin{bmatrix} -(p_1+p_2) & p_3 \\ p_1 & -p_3 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \quad \mathbf{x}(0) = \mathbf{0},$$

$$y_{\text{m}} = [0 \quad 1]\ \mathbf{x},$$

is only locally identifiable, since p_2 and p_3 can be permuted without modifying the input-output behaviour. For any value of $\mathbf{p}$ on the boundary of the posterior feasible set, we know how to generate another value producing the same behaviour, and once a cloud of points in a connected subset of the posterior feasible set is obtained, another can be deduced by symmetry.

In the numerical example treated in (Walter and Piet-Lahanier, 1988), this produced the result shown in Figure 5.21. ◊

EXAMPLE 5.1 (Continued)

Consider again the model

$$y_{\text{m}}(\xi, \mathbf{p}) = p_1\xi_1 + p_2\xi_2 + p_1^3(1 - \xi_1) + p_2^2(1 - \xi_2),$$

An experiment with only two different experimental conditions ξ^1 and ξ^2 makes $\mathbb{S}_{\text{exp}}$ flat, which eliminates parasitic local minimizers. However, identifiability problems may then be introduced, as already observed in Section 4.3.9.1. Assume that the experimental conditions used for three observations are:

$$\xi^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \xi^2 = \xi^3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The two parameter vectors $\mathbf{p} = (p_1, p_2)^{\mathrm{T}}$ and $\mathbf{p}' = (p_1 + 2p_2 - 1,\ 1 - p_2)^{\mathrm{T}}$ then give the same responses $\mathbf{y}^{\mathrm{m}}(\Xi, \mathbf{p})$. When the error bounds are large enough, a connected posterior feasible set is obtained. Conversely, two non-connected sets are obtained for small error bounds. Once a point on the boundary of either of these sets has been obtained, its counterpart on the boundary of the other can be calculated analytically. ◊

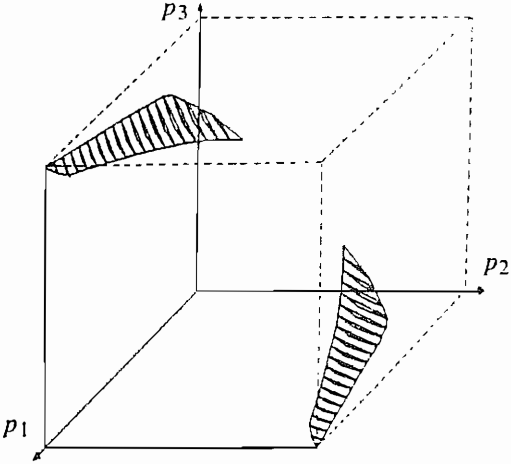


Figure 5.21. Example of a posterior feasible set for a non-uniquely identifiable model

Applying the same technique to the two model structures considered in the chemical engineering example of Section 2.6.4, nine non-connected subsets of the posterior feasible set are obtained (three for one structure, six for the other). PUI's can be estimated by projecting these subsets onto the axes (Walter, Piet-Lahanier and Happel, 1986).

The reasons for the posterior feasible set being not connected may, however, be less easy to detect. It may thus be interesting to use the algorithm mentioned in Section 5.1.3 for characterization of cost contours associated with non-connected domains. Piet-Lahanier and Walter (1990) present a simulated example of a six-parameter globally identifiable structure where four non-connected domains are found, one containing the true value $\mathbf{p}^*$ of the parameters.

5.4.2.3 Set inversion

This approach seems extremely promising, for it yields *global results* even in the case of non-LP model structures. It will only be presented briefly, and the reader is referred to (Moore, 1992; Jaulin and Walter, 1993a, 1993b) for more details. Assume that the posterior feasible set is defined by

$$\mathbb{P}_{\mathrm{m}_t} = \{\mathbf{p} \in \mathbb{P}_0 \mid \mathbf{e}(\mathbf{p}) \in \mathbb{E}\},$$

where $\mathbb{P}_0$ is a prior box denoted by $[\mathbf{p}](0)$, $\mathbf{e}(\mathbf{p}) = \mathbf{y}^{\mathrm{s}} - \mathbf{y}^{\mathrm{m}}(\mathbf{p})$ and $\mathbb{E}$ is a given box in error space. Then

$$\mathbb{P}_{\mathrm{m}_t} = \mathbf{e}^{-1}(\mathbb{E}) \cap \mathbb{P}_0,$$

where $\mathbf{e}^{-1}$ is the inverse function (in a set-theoretic sense) of $\mathbf{e}$. Determining $\mathbb{P}_{\mathrm{m}_t}$ is thus a *set-inversion* problem, which can be solved by the interval-analysis techniques already mentioned in Section 4.3.9.3. The Set Inversion Via Interval Analysis (SIVIA) algorithm described below applies to any function $\mathbf{e}$ for which an inclusion function $\mathbb{e}$ can be calculated. It is not restricted to explicit functions, since inclusion functions can be constructed for differential equations.

To present SIVIA, a few definitions are needed. A *subpaving* of $\mathbb{R}^{n_\mathrm{p}}$ is a set of non-overlapping boxes with non-zero lengths. A box $[\mathbf{p}]$ is said to be *feasible* if $[\mathbf{p}] \subset \mathbb{P}_{\mathrm{m}_t}$, *unfeasible* if $[\mathbf{p}] \cap \mathbb{P}_{\mathrm{m}_t} = \varnothing$ and *ambiguous* otherwise. A *principal hyperplane* of $[\mathbf{p}]$ is one of its hyperplanes of symmetry, orthogonal to an axis with maximal length.

Interval analysis provides two conditions for testing feasibility of any box $[\mathbf{p}]$ in $\mathbb{P}_0$ that will be exploited by the algorithm:

— if $\mathbb{e}([\mathbf{p}]) \subset \mathbb{E}$, then $[\mathbf{p}] \subset \mathbb{P}_{\mathrm{m}_t} \Rightarrow [\mathbf{p}]$ is feasible,
— if $\mathbb{e}([\mathbf{p}]) \cap \mathbb{E} = \varnothing$, then $[\mathbf{p}] \cap \mathbb{P}_{\mathrm{m}_t} = \varnothing \Rightarrow [\mathbf{p}]$ is unfeasible.

In all other cases, $[\mathbf{p}]$ is *indeterminate*, which does not mean it is ambiguous. Figure 5.22 illustrates the various types of boxes considered, together with their images by $\mathbf{e}$ and $\mathbb{e}$.

To store the boxes still to be considered, SIVIA uses a *stack*, *i.e.* a dynamical structure on which only three operations are possible:

— put an element on top,
— remove the element located on top,
— test whether the stack is empty.

We distinguish:

— the subpaving $\mathbb{K}_{\mathrm{in}}$ of all boxes which have been proved to be feasible,
— the subpaving $\mathbb{K}_{\mathrm{i}}$ of all indeterminate boxes with length less than the required precision δ,
— the box $[\mathbf{p}](k)$ considered at iteration k.

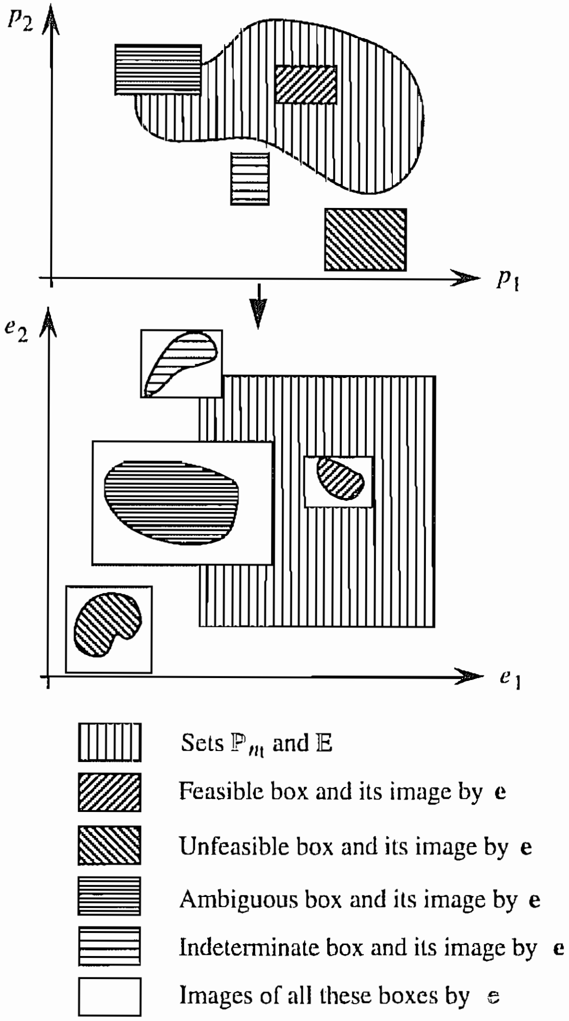

p_2
p_1
e_2
e_1
Sets $\mathbb{P}_m$ and $\mathbb{E}$
Feasible box and its image by **e**
Unfeasible box and its image by **e**
Ambiguous box and its image by **e**
Indeterminate box and its image by **e**
Images of all these boxes by $\mathbb{e}$

Figure 5.22. Boxes and their images by e and its inclusion function $\mathbb{e}$

The basic structure of SIVIA can now be described. The user must supply $\mathbb{E}$, an inclusion function $\mathbb{e}$ for **e**, a prior box $[\mathbf{p}](0)$ in which the search will be performed, and the required precision δ. A prior subpaving could also be used instead of a prior box. The program is initialized by setting:

$$k = 0, \text{ stack} = \emptyset, \mathbb{K}_{\text{in}} = \emptyset, \mathbb{K}_{\text{i}} = \emptyset,$$

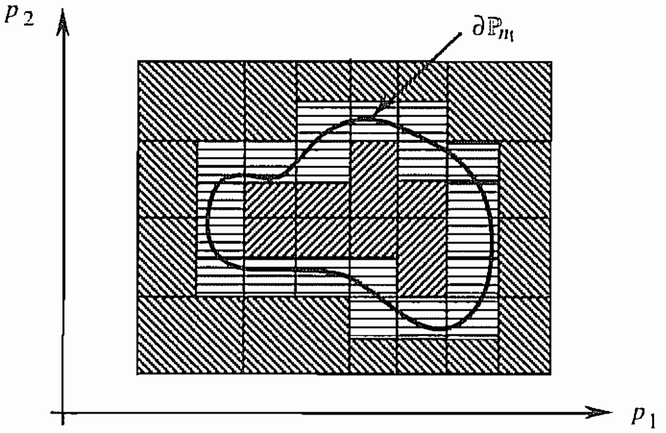
and the kth iteration is:

Step 1: If $\mathbb{e}([\mathbf{p}](k)) \subset \mathbb{E}$, then $\{\mathbb{K}_{\text{in}} = \mathbb{K}_{\text{in}} \cup [\mathbf{p}](k)$. Go to Step 4$\}$.
Step 2: If $\mathbb{e}([\mathbf{p}](k)) \cap \mathbb{E} = \varnothing$, go to Step 4.
Step 3: If $w([\mathbf{p}](k)) \leq \delta$, then $\mathbb{K}_{\text{i}} = \mathbb{K}_{\text{i}} \cup [\mathbf{p}](k)$, else split $[\mathbf{p}](k)$ along a principal hyperplane and put the two resulting boxes on the top of the stack.
Step 4 : If the stack is not empty, remove the element located on top, call it $[\mathbf{p}](k+1)$, increment k by one and go to Step 1.

After running SIVIA, all indeterminate boxes are in $\mathbb{K}_{\text{i}}$ and thus have a length smaller than the precision required. SIVIA encloses $\mathbb{P}_m$ between two subpavings:

$$\mathbb{K}_{\text{in}} \subset \mathbb{P}_m \subset \mathbb{K}_{\text{out}} = \mathbb{K}_{\text{in}} \cup \mathbb{K}_{\text{i}},$$

which tend to coincidence as δ tends to zero (Jaulin and Walter, 1993b). The initial box $[\mathbf{p}](0)$ has thus been partitioned into three subpavings $\mathbb{K}_{\text{in}}$, $\mathbb{K}_{\text{i}}$ and $\mathbb{K}_{\text{un}}$, where $\mathbb{K}_{\text{un}}$ consists of all boxes that have been proved unfeasible (Figure 5.23).



▨ Subpaving $\mathbb{K}_{\text{in}}$ of all boxes that have been proved feasible

☰ Subpaving $\mathbb{K}_{\text{i}}$ of all boxes still indeterminate

▧ Subpaving $\mathbb{K}_{\text{un}}$ of all boxes that have been proved unfeasible

Figure 5.23. Information on $\mathbb{P}_m$ provided by SIVIA

The volume of $\mathbb{K}_{\text{i}}$ (and therefore of the part of $[\mathbf{p}](0)$ for which no conclusion has been reached) can be reduced, at the cost of a reduction in δ that entails an increased number of boxes to be examined. The complexity of SIVIA is analysed in (Jaulin and Walter, 1993a). This algorithm can be extended to compute outlier minimum number estimates (Jaulin, Walter and Didrit, 1996).

5.5 Conclusions

A variety of techniques is available to characterize parameter uncertainty. The method based on the inverse of the Fisher information matrix is very simple to employ, but only justified asymptotically, except in very particular cases (*e.g.*, an LP model structure with additive independent errors distributed $\mathcal{N}(0, \sigma^2)$). The other methods require more intensive computation, but permit characterization of the uncertainty associated with a finite number of data points. All these methods rely on hypotheses that are not necessarily satisfied in practice, so one should be careful in interpreting their results. Note finally that the experiment performed (location of sensors and actuators, shape of the inputs applied, measurement times...) affects the uncertainty in the parameters. It is thus important to choose the experiment well, taking into account its final purpose (for instance, precise estimation of some parameters, or discrimination between model structures). This is experiment design.

6 Experiments

The experimental procedure for the collection of the numerical data to be used to estimate the parameters of a given model depends on *qualitative* choices made at various steps of the modelling. In particular, the tools presented in the previous chapters allow one to choose:

— a set of model structures to be considered,
— location of sensors and actuators to guarantee identifiability (and distinguishability if several rival structures exist),
— an estimator and a characterization of parameter uncertainty.

Once these choices have been made, the experimenter still has some freedom to specify the *quantitative* experimental conditions (such as temperature, pressure, sampling times, shape of inputs...). Experiment design aims at determining experimental conditions adapted to the final purpose of the modelling. It is a crucial step, for a badly designed experiment may ruin any attempt at analyzing the data collected from the system.

We start the presentation with a classical example, which illustrates the benefits of careful experiment design.

EXAMPLE 6.1

Consider three objects O_1, O_2, O_3 with respective weights $\overset{*}{w}_1$, $\overset{*}{w}_2$ and $\overset{*}{w}_3$ to be estimated. The spring balance used for this purpose produces random measurement errors ε, assumed to correspond to independent variables with a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, together with an unknown systematic error $\overset{*}{w}_0$. Four measurements are to be performed.

The simplest approach that comes to mind is first to use the spring balance without any object to estimate the systematic error and then to weigh the three objects successively. Let $y(0)$ be the result of the first measurement, $y(0) = \overset{*}{w}_0 + \varepsilon(0)$. The other measurements give $y(i) = \overset{*}{w}_0 + \overset{*}{w}_i + \varepsilon(i)$, $i = 1, 2, 3$. It is then easy to show that the estimates $\hat{w}_i = y(i) - y(0)$ of the weights $\overset{*}{w}_i$ $(i = 1, 2, 3)$ are unbiased and have variances $\mathrm{var}(\hat{w}_i) = 2\sigma^2$ and covariances $\mathrm{cov}(\hat{w}_i, \hat{w}_j) = \sigma^2$, $i \neq j$.

Consider now another experiment, also with four measurements. It only differs from the previous one in the first weighing, during which the three objects are weighed simultaneously. This yields $y(0) = \overset{*}{w}_0 + \overset{*}{w}_1 + \overset{*}{w}_2 + \overset{*}{w}_3 + \varepsilon(0)$, the last three observations being as previously. The estimates of the weights are then

$$\hat{w}_i = \frac{y(0) + y(i) - y(j) - y(k)}{2}, \quad i = 1, 2, 3, \; i \neq j, \; i \neq k, \; j \neq k.$$

They are unbiased, with $\mathrm{var}(\hat{w}_i) = \sigma^2$ and $\mathrm{cov}(\hat{w}_i, \hat{w}_j) = 0$, $i \neq j$.

Compared to the first intuitive approach, this second experiment thus yields more accurate estimation (with the variances of the estimates reduced by a factor of two) and independent estimates. ◊

The design of an optimal experiment usually consists of the following steps:

— define an optimality criterion related to the final purpose of the modelling, *via* a scalar cost function,
— take into account all constraints on feasible experiments,
— optimize the chosen cost function with respect to the experimental variables available to the experimenter.

Here we shall mainly consider the case of a single model structure, where the purpose is accurate estimation of the parameters of this model. (Experiment design for model discrimination and for the maximization of a model response are briefly considered in Sections 6.6.3 and 4.4.2 respectively.) The optimality criterion will thus be related to how uncertainty in the parameters is characterized, and will depend on the estimator to be used. We notice already that prior knowledge (or hypotheses) about the process should be taken into account to adapt to the specific features of the problem. In a sense, this is unavoidable, since model structures are prior assumptions. When no prior information on the process is available, there is no alternative to a heuristic distribution of experiments over the feasible experimental range.

Assume that the ith scalar observation can be written as $y(\xi^i)$, where the n_ξ-dimensional vector ξ^i (the ith support point) describes the experimental conditions (*e.g.*, measurement time, shape of input...) under which the ith observation is to be collected. When n_t such observations are taken, the concatenation of the vectors ξ^i's yields the $n_t n_\xi$-dimensional vector $\Xi = (\xi^{1\mathrm{T}}, \xi^{2\mathrm{T}}, \ldots, \xi^{n_t\mathrm{T}})^{\mathrm{T}}$, which characterizes all experimental conditions to be optimized. To make experiment design realistic, it is necessary to take a number of constraints into account, *e.g.* on the duration of the experiments, the energy or amplitude of the inputs, the minimum time between samples... Let $\bar{\Xi}$ be the set of all feasible values for Ξ. This set will often have the form $\bar{\Xi} = (\bar{\xi}^{\mathrm{T}}, \ldots, \bar{\xi}^{\mathrm{T}})^{\mathrm{T}}$. However, components of the various ξ^i's may happen not to be independent; see Section 6.3.2.

The definition of a cost function j then permits optimal experiment design to be cast as a constrained optimization problem, where the optimal experiment Ξ^* is defined by

$$\Xi^* = \arg \underset{\Xi \in \bar{\Xi}}{\mathrm{opt}}\; j(\Xi).$$

Statisticians have been interested in optimal experiment design for parameter estimation for many years, and the subject is addressed in several books (Fedorov, 1972; Zarrop, 1979; Silvey, 1980; Penenko, 1981; Ermakov, 1983; Pázman, 1986; Ermakov and Zhigljavsky, 1987; Dodge, Fedorov and Wynn, 1988; Atkinson and Donev, 1992; Pukelsheim, 1993; Schwabe, 1996). A detailed bibliography can be found in (Steinberg and Hunter, 1984; Bandemer, Näther and Pilz, 1987; Rash, 1988; Ford, Kitsos and Titterington, 1989; Walter and Pronzato, 1990). We shall most often assume that the uncertainty in the parameters is characterized by the inverse of the Fisher information matrix (Section 5.3.1). Problems raised by non-LP structures are addressed in Section 6.4. Bayesian estimators are briefly treated in Section 6.5; see also (Chaloner and Verdinelli, 1995). Bounded-errors estimators (Section 5.4) require a specific treatment; see, *e.g.*, (Pronzato and Walter, 1990).

The next section describes various criteria that can be used to design optimal experiments.

6.1 Criteria

Evaluation of the cost function j must be simple enough to allow easy optimization. For that reason, classical optimality criteria correspond to a scalar function of the Fisher information matrix $\mathbf{F}(\mathbf{p}, \Xi)$ presented in Section 5.3.1, *i.e.*

$$j(\Xi) = \phi[\mathbf{F}(\mathbf{p}, \Xi)].$$

Under the hypotheses mentioned in Section 3.3.3, the distribution of the maximum-likelihood estimator is *asymptotically* Gaussian $\mathcal{N}(\mathbf{p}^*, \mathbf{F}^{-1}(\mathbf{p}^*, \Xi))$ when the number of measurements tends to infinity. Minimizing $j(\Xi)$ then amounts to minimizing a scalar measure of the asymptotic covariance matrix of the parameters. Recall, however, that characterizing parameter uncertainty in a non-LP model by the inverse of the Fisher information matrix involves approximations (Section 5.3.1.4), to which few alternatives exist (Sections 5.3.3 and 6.4.1).

A rather general class of optimality criteria employs the following family of cost functions (Kiefer, 1974)

$$\phi_k(\mathbf{F}) = \left[\frac{1}{n_\mathrm{p}} \text{ trace } (\mathbf{Q}\mathbf{F}^{-1}\mathbf{Q}^\mathrm{T})^k\right]^{1/k} \quad \text{if} \quad \det \mathbf{F} \neq 0,$$
$$\phi_k(\mathbf{F}) = \infty \quad \text{if} \quad \det \mathbf{F} = 0,$$

where $\mathbf{Q}$ is a weighting matrix.

The special case $k = 1$ corresponds to the *L-optimality* cost function,

$$j_\mathrm{L}(\Xi) = \text{trace } [\mathbf{Q}^\mathrm{T}\mathbf{Q}\mathbf{F}^{-1}(\mathbf{p}, \Xi)];$$

and choosing $\mathbf{Q} = \mathbf{I}_{n_\mathrm{p}}$ then corresponds to the *A-optimality* cost function. An A-optimal experiment minimizes the sum of the squares of the lengths of the axes of asymptotic confidence ellipsoids. In the context of optimal input design for dynamic systems (Section 6.3.2), it has been suggested (Mehra, 1974a) that trace $\mathbf{F}(\mathbf{p}, \Xi)$ be maximized, which requires less calculation than using the A-optimality cost function (or the D-optimality cost function presented below). However, such an approach may lead to a singular information matrix at the optimum (Grewal and Glover, 1975; Zarrop and Goodwin, 1975), which implies loss of local identifiability of the parameters of the model. Choosing $\mathbf{Q}$ diagonal, with $[\mathbf{Q}]_{ii} = 1/p_i$ corresponds to *C-optimality*, which is connected with the relative precision of estimates. Taking $\mathbf{Q}$ to be a row vector leads to *c-optimality*. Note that the choice of $\mathbf{Q}$ may be dictated by the final purpose of the identification (Goodwin and Payne, 1977).

Taking $\mathbf{Q} = \mathbf{I}_{n_\mathrm{p}}$ and $k = \infty$ corresponds to *E-optimality*; E-optimal design maximizes the smallest eigenvalue of the Fisher information matrix and thus minimizes the length of the largest axis of the asymptotic confidence ellipsoids.

The most widely used optimality criterion has $k = 0$, $\mathbf{Q} = \mathbf{I}_{n_\mathrm{p}}$, requiring minimization of $\det \mathbf{F}^{-1}(\mathbf{p}, \Xi)$, or, equivalently, maximization of

$$j_\mathrm{D}(\Xi) = \det \mathbf{F}(\mathbf{p}, \Xi).$$

An experiment Ξ_D that maximizes j_D is called *D-optimal* (Box and Lucas, 1959). A D-optimal experiment minimizes the volume of the asymptotic confidence ellipsoids for the parameters (Section 6.4.1). Moreover, it is invariant under any nonsingular reparametrization that does not depend on the experiment. Indeed, let $\tilde{\mathbf{p}}(\mathbf{p})$ be such a reparametrization ($\det \partial\tilde{\mathbf{p}}/\partial\mathbf{p}^\mathrm{T} \neq 0$). The D-optimality cost function becomes

$$\det \mathbf{F}(\tilde{\mathbf{p}}, \Xi) = \det [\mathbf{F}(\mathbf{p}, \Xi)] \left[\det \frac{\partial \tilde{\mathbf{p}}}{\partial \mathbf{p}^\mathrm{T}} \right]^{-2}.$$

If the Jacobian $\det \partial\tilde{\mathbf{p}}/\partial\mathbf{p}^\mathrm{T}$ does not depend on the experiment, the maximization of $\det \mathbf{F}(\tilde{\mathbf{p}}, \Xi)$ is equivalent to that of $\det \mathbf{F}(\mathbf{p}, \Xi)$. Note, finally, that a D-optimal experiment generally consists of the repetition of a small number of distinct experimental conditions (*i.e.* some support points ξ_D^i's are equal) (Atkinson and Hunter, 1968; Box, 1968, 1970; Vila, 1988). As mentioned in Section 4.3.9.1, this may help remove parasitic local optimizers when estimating the parameters.

With each of the criteria above can be associated an *efficiency*, which quantifies the suboptimality of an experiment. The D-efficiency of Ξ, for instance, can be defined as

$$j_\mathrm{DE}(\Xi) = \left[\frac{\det \mathbf{F}(\mathbf{p}, \Xi)}{\det \mathbf{F}(\mathbf{p}, \Xi_\mathrm{D})} \right]^{1/n_\mathrm{p}},$$

with Ξ_D a D-optimal experiment. For any Ξ, $j_\mathrm{DE}(\Xi) \leq 1$.

Sometimes, only some of the parameters are of interest. Such will be the case, for instance, when maximum-likelihood estimation of the parameters of a deterministic model requires parameters in the noise distribution to be estimated, as in Example 3.3. Parameters that must be estimated although we are not interested in their values are called *nuisance parameters*. Optimality criteria may then be defined for an accurate estimation of a suitable part of $\mathbf{p}$. Partition $\mathbf{p}$ into $(\mathbf{p}_1^\mathrm{T}, \mathbf{p}_2^\mathrm{T})^\mathrm{T}$, with $\mathbf{p}_1$ the parameters of interest ($\dim \mathbf{p}_1 = s$) and $\mathbf{p}_2$ the nuisance parameters. Partition $\mathbf{F}(\mathbf{p}, \Xi)$ accordingly into

$$\mathbf{F}(\mathbf{p}, \Xi) = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{F}_{22} \end{bmatrix}.$$

$\mathbf{F}^{-1}(\mathbf{p}, \Xi)$, which approximates (sometimes is equal to, see Section 5.3.1) the covariance of the estimator, is then given by

$$\begin{bmatrix} (\mathbf{F}_{11} - \mathbf{F}_{12}\mathbf{F}_{22}^{-1}\mathbf{F}_{21})^{-1} & -(\mathbf{F}_{11} - \mathbf{F}_{12}\mathbf{F}_{22}^{-1}\mathbf{F}_{21})^{-1}\mathbf{F}_{12}\mathbf{F}_{22}^{-1} \\ -\mathbf{F}_{22}^{-1}\mathbf{F}_{21}(\mathbf{F}_{11} - \mathbf{F}_{12}\mathbf{F}_{22}^{-1}\mathbf{F}_{21})^{-1} & \mathbf{F}_{22}^{-1}+\mathbf{F}_{22}^{-1}\mathbf{F}_{21}(\mathbf{F}_{11} - \mathbf{F}_{12}\mathbf{F}_{22}^{-1}\mathbf{F}_{21})^{-1}\mathbf{F}_{12}\mathbf{F}_{22}^{-1} \end{bmatrix},$$

where $\mathbf{F}_{22}$ is assumed not to be singular. Since only $\mathbf{p}_1$ is of interest, we wish to minimize a scalar function of $(\mathbf{F}_{11} - \mathbf{F}_{12}\mathbf{F}_{22}^{-1}\mathbf{F}_{21})^{-1}$. *$D_s$-optimal* design then maximizes

$$j_{\mathrm{D_s}}(\Xi) = \det\,(\mathbf{F}_{11} - \mathbf{F}_{12}\mathbf{F}_{22}^{-1}\mathbf{F}_{21}).$$

It is also used in experiment design for structure discrimination (Section 6.6.3.3). Atwood (1980), Silvey (1980) and Pázman (1986) detail the use of this cost function, in particular when $\mathbf{F}_{22}$ is singular.

REMARK 6.1

Independently of any statistical consideration, one may be interested in choosing an experiment Ξ that makes the approximation $\mathbf{H}_{\mathrm{a}}(\mathbf{p}, \Xi)$ of the Hessian of the estimation cost well conditioned. Vandanjon (1995) has thus considered maximization with respect to Ξ of the Frobenius condition number

$$j_{\mathrm{F}}(\mathbf{p}, \Xi) = \sqrt{\operatorname{trace}\,[\mathbf{H}_{\mathrm{a}}(\mathbf{p}, \Xi)]\,\operatorname{trace}\,[\mathbf{H}_{\mathrm{a}}^{-1}(\mathbf{p}, \Xi)]},$$

with an algorithm similar to the DETMAX algorithm of Section 6.2.1.2. ◊

The initial step of optimal design consists in constructing the Fisher information matrix $\mathbf{F}(\mathbf{p}, \Xi)$. Assume first an additive measurement noise $\varepsilon(t)$ from a sequence of i.i.d. random variables with distribution independent of $\mathbf{p}^*$:

$$y(\xi^i) = y_{\mathrm{m}}(\xi^i, \mathbf{p}^*) + \varepsilon(i),\ i = 1, \dots, n_{\mathrm{t}}.$$

Denote the probability density of ε by π_ε. Using the same development as in Sections 3.7.1 and 5.3.1.4, and under the same hypotheses, the matrix $\mathbf{F}(\mathbf{p}, \Xi)$ associated with the n_{t} observations can be written as

$$\mathbf{F}(\mathbf{p}, \Xi) = \frac{1}{w}\sum_{i=1}^{n_{\mathrm{t}}} \mathbf{s}_{\mathrm{y}}(\xi^i, \mathbf{p})\mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi^i, \mathbf{p}).$$

The weighting term w (inverse of the Fisher information) is

$$w = \left(\int_{\mathbb{D}} \left(\frac{\mathrm{d}\pi_\varepsilon(\varepsilon)}{\mathrm{d}\varepsilon}\right)^2 \frac{1}{\pi_\varepsilon(\varepsilon)}\,\mathrm{d}\varepsilon\right)^{-1},$$

with $\mathbb{D} = \{\varepsilon \mid \pi_\varepsilon(\varepsilon) > 0\}$. For a Gaussian density $\mathcal{N}(0, \sigma^2)$, $w = \sigma^2$. Here $\mathbf{s}_{\mathrm{y}}(\xi^i, \mathbf{p})$ is the sensitivity of the model output with respect to $\mathbf{p}$ (Section 4.3.3.2). When the model structure is LP, $y_{\mathrm{m}}(\xi^i, \mathbf{p})$ is given by $\mathbf{r}(\xi^i)^{\mathrm{T}}\mathbf{p}$, and $\mathbf{s}_{\mathrm{y}}(\xi^i, \mathbf{p})$ is $\mathbf{r}(\xi^i)$. The matrix $\mathbf{F}(\mathbf{p}, \Xi)$ may also be written in the more condensed form

$$\mathbf{F}(\mathbf{p}, \Xi) = \frac{1}{w}\mathbf{S}(\mathbf{p}, \Xi)^{\mathrm{T}}\mathbf{S}(\mathbf{p}, \Xi),$$

where the ith row of the matrix $\mathbf{S}(\mathbf{p}, \Xi)$ is $\mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\boldsymbol{\xi}^i, \mathbf{p})$. Note that the optimal experiment does not depend on the value of w, and thus on the actual density of the measurement noise, which can be assumed Gaussian without loss of generality.

If the distribution of $\varepsilon(i)$ depends on the experimental conditions $\boldsymbol{\xi}^i$, the weighting factor w depends on $\boldsymbol{\xi}^i$, and $\mathbf{F}(\mathbf{p}, \Xi)$ can be written as

$$\mathbf{F}(\mathbf{p}, \Xi) = \sum_{i=1}^{n_t} \frac{1}{w(\boldsymbol{\xi}^i)} \mathbf{s}_{\mathrm{y}}(\boldsymbol{\xi}^i, \mathbf{p})\mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\boldsymbol{\xi}^i, \mathbf{p}).$$

If the measurement noise does not correspond to a sequence of independent variables, instead of considering output errors one should consider the sequence of prediction errors, which is a sequence of independent variables at the true value of the model parameters (Section 3.3.2). The sensitivity of the model output $\mathbf{s}_{\mathrm{y}}(\boldsymbol{\xi}^i, \mathbf{p})$ must then be replaced by the sensitivity of the prediction error. This will be considered in more detail in Section 6.3.2.2.

EXAMPLE 6.1 (continued)

Consider again the determination of the weights of three objects from four measurements. The vector $\mathbf{y}^{\mathrm{s}}$ of observations is

$$\mathbf{y}^{\mathrm{s}} = \mathbf{R}\mathbf{p}^* + \boldsymbol{\varepsilon},$$

with $\mathbf{p}^* = (w_0^*, w_1^*, w_2^*, w_3^*)^{\mathrm{T}}$ and

$$\mathbf{R} = \begin{bmatrix} 1 & r_{01} & r_{02} & r_{03} \\ 1 & r_{11} & r_{12} & r_{13} \\ 1 & r_{21} & r_{22} & r_{23} \\ 1 & r_{31} & r_{32} & r_{33} \end{bmatrix},$$

where $r_{ik} = 1$ or 0 depending on whether the kth object is present on the spring balance for the ith weighing. (The first column of $\mathbf{R}$ only contains ones, since the systematic error w_0^* is always present.) For the two methods suggested above, $r_{ik} = 0$ for $k \neq i$ ($i = 1, 2, 3$). They only differ in the first row of $\mathbf{R}$: $r_{0k} = 0$ for the first method, whereas $r_{0k} = 1$ for the second ($k = 1, 2, 3$). The Fisher information matrix for the estimation of $\mathbf{p}$ is $\mathbf{F} = \mathbf{R}^{\mathrm{T}}\mathbf{R}/\sigma^2$. It does not depend on $\mathbf{p}$ as the structure is LP. Since we are only interested in the values of the parameters w_1, w_2 and w_3, D_{s}-optimality may be used, with w_0 considered a nuisance parameter. ◊

In the previous example, the entries of $\mathbf{R}$ can only take the values zero and one, and the design problem is thus combinatorial, a case which will not be considered in the rest of this chapter. The same kind of problem is met when the purpose of the experiment is *comparison of responses to different treatments* (Searle, 1971; Arnold, 1981; Pearce, 1983). This topic originated in agriculture (Fisher, 1926), hence the traditional vocabulary (blocs, treatments, effects...). A large body of literature is devoted to this subject (Fisher, 1925; Cochran and Cox, 1957; Cox, 1958; Finey, 1960; Box and Draper, 1987; Street, 1987...).

EXAMPLE 6.2

Consider the system

$$y(t) = p_1^* \exp(-p_2^* t) + \varepsilon(t),$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, \sigma^2)$. Assume that two observations $y(t_1)$ and $y(t_2)$ are to be made. The problem is to choose measurement times t_1 and t_2, with $t_2 > t_1 \geq 0$, so as to estimate $\mathbf{p} = (p_1, p_2)^{\mathrm{T}}$ as well as possible. We thus have $\Xi = (t_1, t_2)^{\mathrm{T}}$. The sensitivity of the model response to $\mathbf{p}$ can be written as

$$\mathrm{s_y}(t, \mathbf{p}) = [\exp(-p_2 t), -tp_1 \exp(-p_2 t)]^{\mathrm{T}},$$

which gives the D-optimality cost function

$$j_{\mathrm{D}}(\Xi) = \frac{1}{\sigma^4} p_1^2 (t_2 - t_1)^2 \exp[-2p_2(t_1 + t_2)].$$

The D-optimal experiment is

$$\Xi_{\mathrm{D}} = (0, \frac{1}{p_2})^{\mathrm{T}}. \qquad \Diamond$$

In the example above, Ξ_{D} depends on p_2. The optimal experiment therefore depends on the values of the parameters to be estimated. This problem is common to non-LP structures. The most classical (and simplest) approach then consists of using a nominal value $\mathbf{p}^0$ for the parameters, and designing an optimal experiment for $\mathbf{p} = \mathbf{p}^0$. This is called *local design* and considered in the next section. Approaches that allow uncertainty in $\mathbf{p}^0$ to be taken into account will be presented in Section 6.4.

6.2 Local design

In this section, no distinction will be made between LP and non-LP structures. For non-LP structures, we simply assume that a nominal value $\mathbf{p}^0$ for the parameters has been defined. In some cases, the optimal experiment can be determined analytically; see Example 6.2 above. However, optimization is most often iterative. The type of method to be used will then depend on the dimension of Ξ. When this dimension is not too large, classical nonlinear programming methods, such as those presented in Section 4.3, may be used. Note, however, that the cost function $j(\Xi) = \phi[\mathbf{F}(\mathbf{p}, \Xi)]$ generally has several local optimizers, so a global optimization method is recommended; see Section 4.3.9 and Example 4.22.

When the dimension of Ξ is large, it is preferable to use dedicated algorithms. Their principles are now presented for the case of D-optimal design.

6.2.1 Exact design

Exact means here that optimization is with respect to the variables defining the experiment to be performed. (Compare with the notion of approximate design to be presented in Section 6.2.2.) The most classical algorithms from the literature are *exchange algorithms* (Fedorov, 1972; Mitchell, 1974), for which one of the vectors ξ^i (experimental conditions for the ith observation) is replaced at each iteration by a better vector ξ^* (in the sense of the design criterion). The methods differ in the selection of ξ^i and in the construction of ξ^*. We limit ourselves to presenting their basic ideas.

6.2.1.1 Fedorov's algorithm

Let Ξ^k be the estimate of Ξ at iteration k. Assume Ξ^k is not degenerate (*i.e.* det $\mathbf{F}(\mathbf{p}, \Xi^k) \neq 0$). One of the support points ξ^i of Ξ^k is replaced in Ξ^{k+1} by ξ^*, with ξ^i and ξ^* chosen so that det $\mathbf{F}(\mathbf{p}, \Xi^{k+1}) >$ det $\mathbf{F}(\mathbf{p}, \Xi^k)$. Fedorov (1972) has shown that

$$\frac{j_{\mathrm{D}}(\Xi^{k+1})}{j_{\mathrm{D}}(\Xi^k)} = 1 + \Delta(\xi^i, \Xi^k, \xi^*),$$

with

$$\Delta(\xi, \Xi, \xi^*) = d_1(\xi^*, \Xi) - d_1(\xi, \Xi) + d_2^2(\xi, \Xi, \xi^*) - d_1(\xi^*, \Xi)d_1(\xi, \Xi),$$

where

$$d_1(\xi, \Xi) = \frac{1}{w(\xi)}\, \mathbf{s}_\mathrm{y}^\mathrm{T}(\xi, \mathbf{p})\mathbf{F}^{-1}(\mathbf{p}, \Xi)\mathbf{s}_\mathrm{y}(\xi, \mathbf{p}),$$

and

$$d_2(\xi, \Xi, \xi^*) = \frac{1}{\sqrt{w(\xi)w(\xi^*)}}\, \mathbf{s}_\mathrm{y}^\mathrm{T}(\xi, \mathbf{p})\mathbf{F}^{-1}(\mathbf{p}, \Xi)\mathbf{s}_\mathrm{y}(\xi^*, \mathbf{p}).$$

The algorithm is then as follows:

Step 1: Choose some non-degenerate Ξ^1, $\delta << 1$ and set $k = 1$.
Step 2: Find

$$(\xi^{i*}, \xi^*) = \arg \max_{\xi^i \in \operatorname{supp}\{\Xi^k\}} \max_{\xi \in \boldsymbol{\xi}} \Delta(\xi^i, \Xi^k, \xi)$$

where supp$\{\Xi\}$ denotes the set of support points of Ξ, and $\boldsymbol{\xi}$ is the admissible set for ξ.
Step 3: If $\Delta(\xi^{i*}, \Xi^k, \xi^*) < \delta$, stop. Otherwise, replace ξ^{i*} by ξ^*, increment k by one and go to Step 2.

REMARKS 6.2

— At each step, computing $\Delta(\xi^i, \Xi^k, \xi)$ requires evaluation of $\mathbf{F}^{-1}(\mathbf{p}, \Xi^k)$. Using the matrix-inversion lemma, one can show that

$$\mathbf{F}^{-1}(\mathbf{p}, \Xi^{k+1}) = \{\mathbf{I}_{n_\mathrm{p}} - \mathbf{F}^{-1}(\mathbf{p}, \Xi^k)\mathbf{M}[\mathbf{I}_2 + \mathbf{M}^\mathrm{T}\mathbf{F}^{-1}(\mathbf{p}, \Xi^k)\mathbf{M}]^{-1}\mathbf{M}^\mathrm{T}\}\mathbf{F}^{-1}(\mathbf{p}, \Xi^k),$$

with

$$\mathbf{M} = \left[\; j\frac{\mathbf{s}_y(\xi^{i*}, \mathbf{p})}{\sqrt{w(\xi^{i*})}} \quad \frac{\mathbf{s}_y(\xi^{*}, \mathbf{p})}{\sqrt{w(\xi^{*})}} \; \right]$$

and $j^2 = -1$. Updating $\mathbf{F}^{-1}$ thus only requires inversion of a 2×2 matrix.

— A similar algorithm, with another function $\Delta(\xi, \Xi, \xi^*)$, can be used for L-optimal (exact) design (Fedorov, 1972). ◊

EXAMPLE 6.3

Consider the system

$$y(t) = p_1^* \exp(-p_2^* t) + p_3^* \exp(-p_4^* t) + \varepsilon(t),$$

with the $\varepsilon(t)$'s i.i.d. $\mathcal{N}(0,\sigma^2)$. Assume that twelve observations are to be performed to estimate $\mathbf{p} = (p_1, p_2, p_3, p_4)^{\mathrm{T}}$. The problem is to determine the vector $\mathbf{t}_{\mathrm{D}}$ of the twelve D-optimal sampling times. The parameters p_1 and p_3 appear linearly in the model response, and thus have no influence on $\mathbf{t}_{\mathrm{D}}$. Assume that the prior values for p_2 and p_4 are $p_2 = 1$ and $p_4 = 0.1$. With an admissible set $\underset{\sim}{\xi}$ obtained by discretization of the interval [0, 20] with step 0.1, *i.e.*

$$\underset{\sim}{\xi} = \{0, 0.1, 0.2, \ldots , 19.9, 20\},$$

Fedorov's algorithm initialized at

$$\Xi^1 = \mathbf{t}^1 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1., 1.2, 1.5, 10)^{\mathrm{T}}$$

converges in 19 iterations to

$$\mathbf{t}_{\mathrm{D}} = (0, 0, 0, 0.9, 0.9, 0.9, 3.8, 3.8, 3.8, 14.4, 14.4, 14.4)^{\mathrm{T}},$$

with $\det[\mathbf{F}(\mathbf{p}, \mathbf{t}_{\mathrm{D}})]/\det[\mathbf{F}(\mathbf{p}, \mathbf{t}^1)] = 54.58$. ◊

At each iteration k, Fedorov's algorithm requires solution of

$$\max_{\xi \in \underset{\sim}{\xi}} \Delta(\xi^i, \Xi^k, \xi)$$

at each support point ξ^i of Ξ^k. These n_t maximizations of a possibly multimodal function make computation heavy. Moreover, only one support point is modified at Step 2, which may lead to a deadlock when only the simultaneous modification of several would permit improvement of the cost. The next algorithm aims at overcoming these two limitations.

6.2.1.2 DETMAX algorithm

Assume that Ξ^k is not degenerate. Should an additional observation be allowed, characterized by ξ^{n_t+1}, one would choose $\xi^{n_t+1} = \xi^*$, with ξ^* such that $\det \mathbf{F}(\mathbf{p}, \Xi^{k+})$ be maximized, where

$$\Xi^{k+} = \begin{bmatrix} \Xi^k \\ \xi^{n_t+1} \end{bmatrix}.$$

If ξ^{n_t+1} were to replace one of the support points of Ξ^k, one should obtain Ξ^{k+1} from Ξ^{k+} by removing ξ^{i*} such that $\det \mathbf{F}(\mathbf{p}, \Xi^{k+1})$ remains as large as possible. This augmentation of the number of support points, followed by the removal of some of them so as to keep the number of observations equal to n_t, is called an *excursion*. If n_t is strictly larger than the number of parameters n_p, the ordering of the operations may be reversed, and a support point may be removed before the introduction of a new one. This ordering may also be chosen randomly at each iteration.

When the number of support points of Ξ only varies by one, the excursions are said to be of length one. The algorithm DETMAX is then as follows (Mitchell, 1974):

Step 1: Choose some non-degenerate Ξ^1, and set $k = 1$.
Step 2: Find

$$\xi^* = \arg \max_{\xi \in \xi} \frac{1}{w(\xi)} \mathbf{s}_y^T(\xi, \mathbf{p}) \mathbf{F}^{-1}(\mathbf{p}, \Xi^k) \mathbf{s}_y(\xi, \mathbf{p}).$$

Update $\mathbf{F}^{-1}$ for the experiment

$$\Xi^{k+} = \begin{bmatrix} \Xi^k \\ \xi^* \end{bmatrix}$$

according to

$$\mathbf{F}^{-1}(\mathbf{p}, \Xi^{k+}) = \mathbf{F}^{-1}(\mathbf{p}, \Xi^k) - \frac{\mathbf{F}^{-1}(\mathbf{p}, \Xi^k)\mathbf{s}_y(\xi^*, \mathbf{p})\mathbf{s}_y^T(\xi^*, \mathbf{p})\mathbf{F}^{-1}(\mathbf{p}, \Xi^k)}{w(\xi^*) + \mathbf{s}_y^T(\xi^*, \mathbf{p})\mathbf{F}^{-1}(\mathbf{p}, \Xi^k)\mathbf{s}_y(\xi^*, \mathbf{p})}.$$

Step 3: Find

$$\xi^{i*} = \arg \min_{\xi^i \in \operatorname{supp}\{\Xi^{k+}\}} \frac{1}{w(\xi^i)} \mathbf{s}_y^T(\xi^i, \mathbf{p}) \mathbf{F}^{-1}(\mathbf{p}, \Xi^{k+}) \mathbf{s}_y(\xi^i, \mathbf{p}).$$

Step 4: If $\xi^{i*} = \xi^*$, stop.
Step 5: Remove ξ^{i*} from Ξ^{k+} to get Ξ^{k+1}, update $\mathbf{F}^{-1}$ according to

$$\mathbf{F}^{-1}(\mathbf{p}, \Xi^{k+1}) = \mathbf{F}^{-1}(\mathbf{p}, \Xi^{k+}) + \frac{\mathbf{F}^{-1}(\mathbf{p}, \Xi^{k+})\mathbf{s}_y(\xi^{i*}, \mathbf{p})\mathbf{s}_y^T(\xi^{i*}, \mathbf{p})\mathbf{F}^{-1}(\mathbf{p}, \Xi^{k+})}{w(\xi^{i*}) - \mathbf{s}_y^T(\xi^{i*}, \mathbf{p})\mathbf{F}^{-1}(\mathbf{p}, \Xi^{k+})\mathbf{s}_y(\xi^{i*}, \mathbf{p})},$$

increment k by one and go to Step 2.

When excursions of length $\lambda \geq 2$ are used, the algorithm can be summarized as follows. Addition of λ support points is by performing Step 2 λ times. Removing λ support points corresponds to performing Steps 3 and 5 λ times. The stopping test at Step 4 is replaced by a test on the increase of det $\mathbf{F}$. With larger λ, the increase in computing time may not by compensated by the improvement in det $\mathbf{F}$. Mitchell (1974) suggests use of a variable excursion length λ, with $\lambda \leq 6$ to keep the amount of computation reasonable.

Convergence to the optimal experiment is not guaranteed by algorithms for exact design, although they can only increase the value of $j_{\mathrm{D}}(\Xi)$. It is thus recommended that several optimizations be performed, with different initializations Ξ^1 at Step 1, for instance randomly generated in $\bar{\Xi}$. If the computational time is not a limitation, a global optimization algorithm can also be used (Example 4.22).

Various improvements of the algorithms above have been suggested (Atkinson and Donev, 1989; Cook and Nachtsheim, 1980; Galil and Kiefer, 1980; Johnson and Nachtsheim, 1983). The improvements, sometimes quite significant, generally result from intuitive ideas. However, convergence to the optimal experiment is still not guaranteed, and numerical examples are the only basis for a comparison between methods.

In contrast to algorithms for exact design, the methods presented in the next section yield a global optimum, sometimes at the price of approximation to make the designed experiment implementable.

6.2.2 Distribution of experimental effort

6.2.2.1 Continuous design

Each observation $y(\xi^i)$ depends on experimental conditions $\xi^i \in \xi$. When some experiments are repeated, the number n_c of distinct ξ^i's is less than the total number of observations n_t. The Fisher information matrix can then be written as

$$\mathbf{F}(\mathbf{p}, \Xi) = \sum_{i=1}^{n_c} r_i \frac{1}{w(\xi^i)} \mathbf{s}_y(\xi^i, \mathbf{p}) \mathbf{s}_y^{\mathrm{T}}(\xi^i, \mathbf{p}),$$

with r_i the number of repetitions of measurements under the experimental conditions ξ^i, so

$$\sum_{i=1}^{n_c} r_i = n_t.$$

Let $\mathbf{F}_{\mathrm{ps}}(\mathbf{p}, \Xi)$ be the Fisher information matrix per sample;

$$\mathbf{F}_{\mathrm{ps}}(\mathbf{p}, \Xi) = \sum_{i=1}^{n_c} \frac{r_i}{n_t} \frac{1}{w(\xi^i)} \mathbf{s}_y(\xi^i, \mathbf{p}) \mathbf{s}_y^{\mathrm{T}}(\xi^i, \mathbf{p}).$$

The proportion $\mu_i = r_i/n_t$ of observations performed at ξ^i can be considered as the *percentage of experimental effort* spent at ξ^i. The experiment Ξ can thus be represented as a discrete distribution m_D,

$$m_D = \begin{Bmatrix} \xi^1 & \dots & \xi^{n_e} \\ \mu_1 & \dots & \mu_{n_e} \end{Bmatrix},$$

which is normalized, since

$$\sum_{i=1}^{n_e} \mu_i = 1.$$

The μ_i's are rational numbers ($\mu_i = r_i/n_t$, with n_t fixed). Removing this constraint, we can think of an experiment as a probability distribution on $\mathbb{\xi}$.

Chernoff (1953) first expressed the Fisher information matrix per sample as a normalized linear combination of rank-one matrices of the type $\mathbf{s}_y(\xi^i, \mathbf{p})\mathbf{s}_y^T(\xi^i, \mathbf{p})/w(\xi^i)$. Kiefer and Wolfowitz (1959) showed that extending the notion of design to any normalized measure on $\mathbb{\xi}$ (*design measure*) drastically simplifies design. This idea is the basis for the algorithms to be presented in Section 6.2.2.4.

Consider a normalized measure m on $\mathbb{\xi}$, satisfying

$$\int_{\mathbb{\xi}} m(\mathrm{d}\xi) = 1,$$

with $m(\mathrm{d}\xi)$ the measure of the elementary part $\mathrm{d}\xi$ around ξ. The notion of experimental design can be extended to measures on $\mathbb{\xi}$ absolutely continuous with respect to the Lebesgue measure, hence the name *continuous* design. The matrix $\mathbf{F}_{ps}(\mathbf{p}, m)$ then takes the form

$$\mathbf{F}_{ps}(\mathbf{p}, m) = \int_{\mathbb{\xi}} \frac{1}{w(\xi)} \mathbf{s}_y(\xi, \mathbf{p})\mathbf{s}_y^T(\xi, \mathbf{p})\ m(\mathrm{d}\xi).$$

As explained in the next section, any matrix of this form can also be obtained with a discrete measure.

6.2.2.2 Approximate design

$\mathbf{F}_{ps}(\mathbf{p}, m)$ belongs to the convex hull of the rank-one matrices of the type $\mathbf{s}_y(\xi, \mathbf{p})\mathbf{s}_y^T(\xi, \mathbf{p})/w(\xi)$. Moreover, $\mathbf{F}_{ps}(\mathbf{p}, m)$ is symmetric and thus belongs to a $[n_p(n_p + 1)/2]$-dimensional space. Caratheodory's theorem (Berger, 1979, 1987), then indicates that $\mathbf{F}_{ps}(\mathbf{p}, m)$ can always be written as a linear combination of at most $[n_p(n_p + 1)/2] + 1$ rank-one matrices, *i.e.*

$$\mathbf{F}_{ps}(\mathbf{p}, m) = \sum_{i=1}^{n_e} \mu_i \frac{1}{w(\xi^i)} \mathbf{s}_y(\xi^i, \mathbf{p})\mathbf{s}_y^T(\xi^i, \mathbf{p}),$$

with

$$\sum_{i=1}^{n_e} \mu_i = 1 \quad \text{and} \quad n_e \le \frac{n_p(n_p + 1)}{2} + 1.$$

Any continuous design measure is thus equivalent to a discrete design measure m_d with at most $[n_p(n_p + 1)/2] + 1$ support points ξ^i with weights μ_i. This is true, in particular, for the optimal design for any given cost function $\phi[\mathbf{F}_{ps}(\mathbf{p}, m)]$. For D-optimality, m_D that maximizes det $\mathbf{F}_{ps}(\mathbf{p}, m)$ only needs $n_p(n_p + 1)/2$ support points at most, because $\mathbf{F}_{ps}(\mathbf{p}, m_D)$ lies on the boundary of the convex hull of the set of rank-one matrices of the type $\mathbf{s}_y(\xi, \mathbf{p})\mathbf{s}_y^T(\xi, \mathbf{p})/w(\xi)$ (Silvey, 1980).

We can thus restrict our attention to *discrete* experiments, associated with normalized discrete measures on ξ. However, a discrete experiment usually cannot be implemented exactly for a given number n_t of observations, since the weights must satisfy $\mu_i = r_i/n_t$. In most cases, the μ_i's must be approximated by rationals r_i/n_t (Pukelsheim and Rieder, 1992). For that reason, this approach is called *approximate* design.

REMARK 6.3

A discrete experiment is sometimes implementable without any approximation. This will be the case for instance for the design of optimal inputs, when the input signal is characterized by its power spectral density considered as a density of experimental effort (Section 6.3.2.2). ◊

We shall now consider the main properties related to this formulation of the design problem.

6.2.2.3 Properties of optimal experiments

Kiefer-Wolfowitz equivalence theorem. The cost function ϕ is generally chosen to be convex or concave, depending on whether it must be minimized or maximized. For D-optimality, det $\mathbf{F}^{-1}$ has to be minimized, *i.e.* det $\mathbf{F}$ is maximized, or rather ln det $\mathbf{F}$, which is concave on the set of symmetric non-negative definite matrices: for any such matrices $\mathbf{F}_1$, $\mathbf{F}_2$, with $\mathbf{F}_1 \neq \mathbf{F}_2$, and any scalar α such that $0 < \alpha < 1$,

$$\ln \det [(1 - \alpha)\mathbf{F}_1 + \alpha\mathbf{F}_2] > (1 - \alpha) \ln \det \mathbf{F}_1 + \alpha \ln \det \mathbf{F}_2.$$

The set of matrices $\mathbf{F}_{ps}(\mathbf{p}, m)$ is convex, and the design problem amounts to minimization of a convex function ϕ over a convex set. The optimum is thus unique, in the sense that the matrix $\mathbf{F}_{ps}(\mathbf{p}, m)$ associated with an optimal design measure is unique. This does not imply uniqueness of the optimal design measure. However, the set of optimal design measures is convex. The optimum can be characterized by first-order stationarity conditions. What follows concerns D-optimality, but similar results exist for other criteria with suitable convexity properties (Kiefer, 1974).

The design measure m_1 is D-optimal if and only if det $\mathbf{F}_{ps}(\mathbf{p}, m_1)$ is a maximum, or, equivalently, if and only if for any measure m_2,

$$\frac{\partial \ln \det \mathbf{F}_{ps}[\mathbf{p}, (1 - \alpha)m_1 + \alpha m_2]}{\partial \alpha}\bigg|_{\alpha=0} \le 0.$$

Since

$$\frac{\partial \ln \det \mathbf{F}}{\partial \alpha} = \operatorname{trace}\left(\mathbf{F}^{-1}\frac{\partial \mathbf{F}}{\partial \alpha}\right),$$

a necessary and sufficient condition for D-optimality of m_1 is

$$\operatorname{trace}\left\{\mathbf{F}_{\mathrm{ps}}^{-1}(\mathbf{p}, m_1)[\mathbf{F}_{\mathrm{ps}}(\mathbf{p}, m_2) - \mathbf{F}_{\mathrm{ps}}(\mathbf{p}, m_1)]\right\} \leq 0, \text{ for any } m_2,$$

or equivalently

$$\operatorname{trace}\left[\mathbf{F}_{\mathrm{ps}}^{-1}(\mathbf{p}, m_1)\mathbf{F}_{\mathrm{ps}}(\mathbf{p}, m_2)\right] \leq n_{\mathrm{p}}, \text{ for any } m_2.$$

In particular, this must be true when m_2 is the discrete measure with unit weight at a single support point $\xi \in \Xi$. A necessary condition for D-optimality of m_1 is thus

$$d(\xi, m_1) \leq n_{\mathrm{p}}, \text{ for any } \xi \in \Xi,$$

with

$$d(\xi, m) = \frac{1}{w(\xi)}\, \mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi, \mathbf{p})\mathbf{F}_{\mathrm{ps}}^{-1}(\mathbf{p}, m)\mathbf{s}_{\mathrm{y}}(\xi, \mathbf{p}).$$

On the other hand, since $\mathbf{F}_{\mathrm{ps}}(\mathbf{p}, m_2)$ can always be written as a linear combination of at most $[n_{\mathrm{p}}(n_{\mathrm{p}} + 1)/2] + 1$ matrices of the type $\mathbf{s}_{\mathrm{y}}(\xi, \mathbf{p})\mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi, \mathbf{p})/w(\xi)$, one can easily check that this necessary condition is also sufficient. Now, using the fact that

$$n_{\mathrm{p}} = \operatorname{trace}\left[\mathbf{F}_{\mathrm{ps}}^{-1}(\mathbf{p}, m_1)\mathbf{F}_{\mathrm{ps}}(\mathbf{p}, m_1)\right] = \sum_{i=1}^{n_{\mathrm{e}}} \mu_i d(\xi^i, m_1),$$

where the μ_i's and ξ^i's are respectively the weights and support points of the D-optimal design measure m_1, the following theorem is obtained.

EQUIVALENCE THEOREM (Kiefer and Wolfowitz, 1960)

The following properties are equivalent:

— the design measure m_{D} is D-optimal,
— $\max_{\xi \in \Xi} d(\xi, m_{\mathrm{D}}) = n_{\mathrm{p}}$,
— m_{D} minimizes $\max_{\xi \in \Xi} d(\xi, m)$. ◊

If $w(\xi)$ is constant, for instance equal to σ^2 for measurement errors i.i.d. $\mathcal{N}(0, \sigma^2)$, $d(\xi, m)$ is proportional to the variance of the predicted model response at ξ (obtained by linear approximation for a non-LP structure). A D-optimal design measure thus minimizes the maximum of the prediction variance with respect to $\xi \in \Xi$, which corresponds to *G-optimality*. This equivalence between D- and G-optimality does not hold true for exact design. G-optimal design for heteroscedastic models ($w(\xi)$ not constant) is considered in (Wong and Cook, 1993), where an optimization algorithm is suggested.

One can show (Sibson, 1972; Silvey, 1980) that the determination of an approximate D-optimal design is equivalent to finding the minimum-volume ellipsoid centred at the origin and containing the sensitivity set

$$\mathbb{S} = \{\mathbf{s}_y(\xi, \mathbf{p})/\sqrt{w(\xi)} \mid \xi \in \xi\}.$$

The support points of m_D correspond to the contact points of this ellipsoid with $\mathbb{S}$. When their number is n_p, the optimal weights are all $1/n_p$. ◊

EXAMPLE 6.4

Consider the system

$$y(t) = \frac{p_1^*}{p_1^* - p_2^*}[\exp(-p_1^* t) - \exp(-p_2^* t)] + \varepsilon(t),$$

with the measurement errors $\varepsilon(t)$ assumed to be i.i.d. $\mathcal{N}(0, \sigma^2)$, with $\sigma^2 = 1$. A D-optimal experiment (choice of sampling times) is to be determined for nominal parameter values $\mathbf{p} = (0.7, 0.2)^T$. Figure 6.1 presents the sensitivity set

$$\mathbb{S} = [\mathbf{s}_y(t, \mathbf{p})/\sigma \mid t \in [0, 3]\}$$

and the minimum-volume ellipsoid centred at $\mathbf{0}$ containing it.

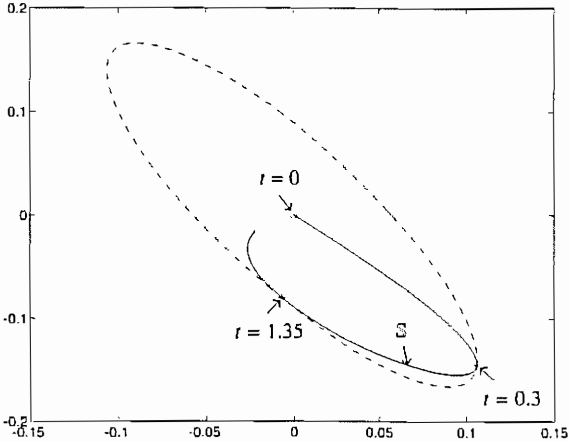


Figure 6.1. Sensitivity set (solid line) for t between 0 and 3, and minimum-volume ellipsoid centred at $\mathbf{0}$ containing it for Example 6.4; the contact points give the support points of m_D

The contact points, indicated by crosses, correspond to $t_1 \approx 0.3$ and $t_2 \approx 1.35$, which are the support points of the D-optimal design measure. Since $n_p = 2$, each point receives weight 1/2. ◊

EXAMPLE 6.5

Consider the same system as in Example 4.21

$$y(\xi) = p_1^* \xi_1 + p_2^* \xi_2 + p_1^{*3}(1-\xi_1) + p_2^{*2}(1-\xi_2) + \varepsilon(\xi),$$

with $\varepsilon(\xi)$ assumed to correspond to an i.i.d. sequence $\mathcal{N}(0, \sigma^2)$, with $\sigma^2 = 1$. The sensitivity $\mathbf{s}_y(\xi, \mathbf{p})$ is

$$\mathbf{s}_y(\xi, \mathbf{p}) = \begin{bmatrix} \xi_1 + 3p_1^2(1-\xi_1) \\ \xi_2 + 2p_2(1-\xi_2) \end{bmatrix}.$$

Assume a nominal value $\mathbf{p} = (1, 2)^T$, and an admissible design space defined by

$$\Xi = \{\xi = (\xi_1, \xi_2)^T \mid \xi_1 \in [0, 1], \xi_2 \in [0, 1]\}.$$

The sensitivity set $\mathbb{S}$ is then the orthotope $[1, 3] \times [1, 4]$. The minimum-volume ellipsoid centred at the origin and containing $\mathbb{S}$ touches $\mathbb{S}$ at three vertices, namely $(3, 4)^T$, $(3, 1)^T$, $(1, 4)^T$, obtained respectively for $\xi_D^1 = (0, 0)^T$, $\xi_D^2 = (0, 1)^T$ and $\xi_D^3 = (1, 0)^T$. Since there are three support points for only two parameters, the optimal weights are not 1/3. Their exact values, determined from the stationarity condition for $\det \mathbf{F}_{ps}(\mathbf{p}, m_D)$, are $\mu_1 = 121/840$, $\mu_2 = 46/105$ and $\mu_3 = 117/280$. Figure 6.2 presents the confidence regions $\mathbb{R}_1^{0.05}$ when $\sigma^2 = 0.25$.

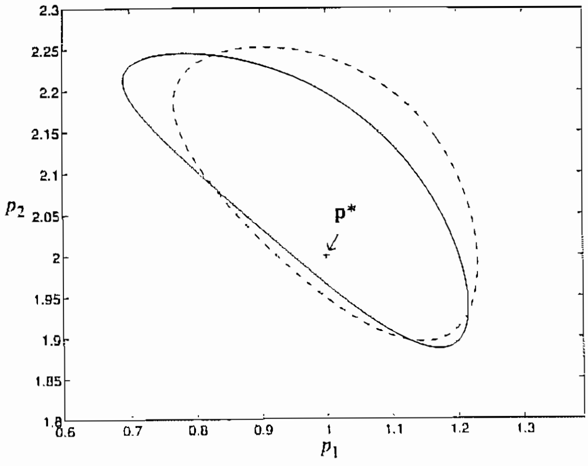


Figure 6.2. Confidence regions $\mathbb{R}_1^{0.05}$ for Example 6.5

The dashed line corresponds to the experiment $\xi^1 = (1, 1)^T$, $\xi^2 = (0, 1)^T$ and $\xi^3 = (1, 0)^T$, with $y(\xi^1) = 3.3354$, $y(\xi^2) = 3.1818$ and $y(\xi^3) = 5.3303$, simulated by adding random errors to the model response at $\mathbf{p}^* = (1, 2)^T$. The solid line corresponds to three observations performed at the support points of the D-optimal experiment. The same errors have been added to the model responses, yielding the observed values $y(\xi_{\mathrm{D}}^1) = 5.3354$, $y(\xi_{\mathrm{D}}^2) = 3.1818$ and $y(\xi_{\mathrm{D}}^3) = 5.3303$. Although the regions are not quite ellipsoidal (Section 6.4.1), choosing the support points of the D-optimal experiment produces a smaller confidence region. The numerical evaluation of their volumes by triangulation gives the values 10.7×10^{-2} for the dashed-line region, and 9.2×10^{-2} for the solid-line region. The improvement might of course have been much greater for other choices of the initial experiment. Note that the initial experiment has two support points in common with the D-optimal experiment. ◊

EXAMPLE 6.2 (continued)

Consider again the system

$$y(t) = p_1^* \exp(-p_2^* t) + \varepsilon(t),$$

with the $\varepsilon(t)$'s i.i.d. $\mathcal{N}(0, \sigma^2)$. The number of observations to be made is not specified, and a D-optimal design measure is sought, with an admissible domain $[0, \infty[$ for the measurement times. It can readily be checked that the design measure

$$m_{\mathrm{D}} = \left\{ \begin{array}{cc} 0 & 1/p_2 \\ 1/2 & 1/2 \end{array} \right\}$$

is D-optimal. Indeed,

$$\mathbf{F}_{\mathrm{ps}}(\mathbf{p}, m_{\mathrm{D}}) = \frac{1}{2\sigma^2} \begin{bmatrix} 1 + 1/e^2 & -p_1/(p_2 e^2) \\ -p_1/(p_2 e^2) & p_1^2/(p_2^2 e^2) \end{bmatrix},$$

and

$$d(t, m_{\mathrm{D}}) = 2 \exp(-2p_2 t)\,(1 - 2p_2 t + p_2^2 t^2 e^2 + p_2^2 t^2)\,.$$

Figure 6.3 presents the evolution of $d(t, m_{\mathrm{D}})$ as t varies between 0 and 2, for $p_2 = 2$. Since $d(0, m_{\mathrm{D}}) = d(1/p_2, m_{\mathrm{D}}) = 2$, and $d(t, m_{\mathrm{D}}) \leq 2$ for all $t \geq 0$, m_{D} is D-optimal. Compare with Figure 6.4, which shows an example of the evolution of $d(t, m)$ when m is not D-optimal. ◊

D-optimality for sums and products of models. The equivalence theorem above allows the following properties for D-optimal design to be proved. Consider two LP structures given by

$$y_{\mathrm{m}1}[\xi_1, \mathbf{p}^{(1)}] = \sum_{i=1}^{n_1} p_i^{(1)} f_i^{(1)}(\xi_1), \text{ with } f_1^{(1)}(\xi_1) = 1, \xi_1 \in \Xi_1$$

and

$$y_{\mathrm{m}2}[\xi_2, \mathbf{p}^{(2)}] = \sum_{i=1}^{n_2} p_i^{(2)} f_i^{(2)}(\xi_2), \text{ with } f_1^{(2)}(\xi_2) = 1, \xi_2 \in \Xi_2,$$

and let m_1 and m_2 be two design measures, D-optimal for $y_{\mathrm{m}1}$ and $y_{\mathrm{m}2}$ respectively.

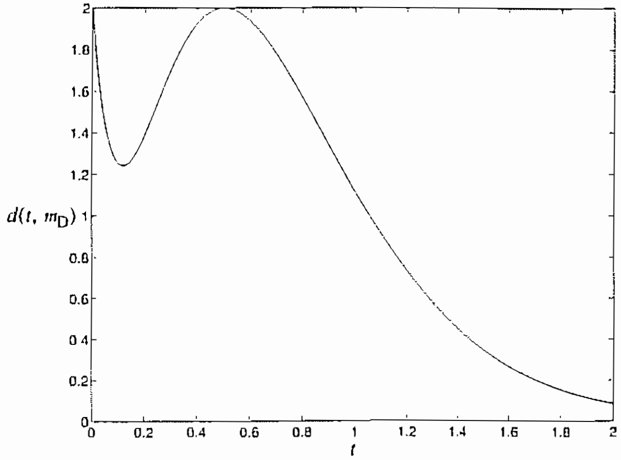


Figure 6.3. Evolution of $d(t, m_D)$ in Example 6.2 as t varies between 0 and 2

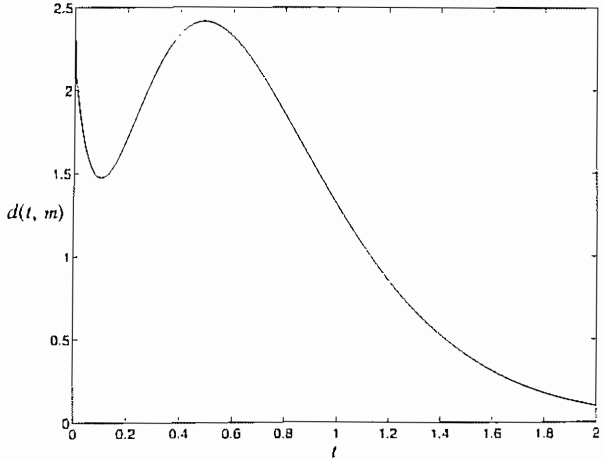


Figure 6.4. Evolution of $d(t, m)$ in Example 6.2 as t varies between 0 and 2, when m is not D-optimal

The product measure $m_1 \otimes m_2$ on $\xi_1 \times \xi_2$ is D-optimal for the *sum of structures*

$$y_{m_s}(\xi_1, \xi_2, \mathbf{p}) = p_1 + p_2 f_2^{(1)}(\xi_1) + \ldots + p_{n_1} f_{n_1}^{(1)}(\xi_1)$$
$$+ p_{n_1+1} f_2^{(2)}(\xi_2) + \ldots + p_{n_1+n_2-1} f_{n_2}^{(2)}(\xi_2),$$

with dim $\mathbf{p} = n_1 + n_2 - 1$, and for the *product of structures*

$$y_{\mathrm{mp}}(\xi_1, \xi_2, \mathbf{p}) = \sum_{i,k} p_{ik} f_i^{(1)}(\xi_1) f_k^{(2)}(\xi_2),$$

with dim $\mathbf{p} = n_1 n_2$.

These properties may permit the design problem for a complicated model structure to be split into design problems for simpler structures. They are often useful for models polynomial in $\boldsymbol{\xi}$, such as those used with the response-surface methodology (Section 4.4.2). That a similar property may also be valid for the sum of non-LP model structures is even more interesting (Schwabe, 1995).

EXAMPLE 6.6

Consider the structure

$$y_{\mathrm{m}}(\boldsymbol{\xi}, \mathbf{p}) = p_0 + p_1\xi_1 + p_2\xi_2 + p_3\xi_1^2 + p_4\xi_2^2,$$

with $\boldsymbol{\xi} = (\xi_1, \xi_2)^{\mathrm{T}}$, $\xi_1 \in [-1, 1]$, $\xi_2 \in [-1, 1]$. It corresponds to the sum of $y_{\mathrm{m1}}[\xi_1, \mathbf{p}^{(1)}]$ and $y_{\mathrm{m2}}[\xi_2, \mathbf{p}^{(2)}]$, with

$$y_{\mathrm{m1}}[\xi_1, \mathbf{p}^{(1)}] = p_0^{(1)} + p_1^{(1)}\xi_1 + p_2^{(1)}\xi_1^2$$

and

$$y_{\mathrm{m2}}[\xi_2, \mathbf{p}^{(2)}] = p_0^{(2)} + p_1^{(2)}\xi_2 + p_2^{(2)}\xi_2^2.$$

The D-optimal design measures for y_{m1} and y_{m2} coincide and are given by

$$m_{\mathrm{D1}} = m_{\mathrm{D2}} = \left\{ \begin{matrix} -1 & 0 & 1 \\ 1/3 & 1/3 & 1/3 \end{matrix} \right\}.$$

The D-optimal design measure for the sum of structures y_{m} is thus

$$m_{\mathrm{D}} = \left\{ \begin{matrix} \begin{bmatrix} -1 \\ -1 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \end{bmatrix} & \begin{bmatrix} -1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ -1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ -1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 \end{matrix} \right\}.$$

It is also D-optimal for the product structure

$$\begin{aligned} y_{\mathrm{mp}}(\boldsymbol{\xi}, \mathbf{p}) = {} & p_0 + p_1\xi_1 + p_2\xi_2 + p_3\xi_1^2 + p_4\xi_2^2 + p_5\xi_1\xi_2 \\ & + p_6\xi_1^2\xi_2 + p_7\xi_1\xi_2^2 + p_8\xi_1^2\xi_2^2. \end{aligned}$$

◊

6.2.2.4 Algorithms

Fedorov-Wynn algorithm. Consider a non-degenerate design measure m^k. From the equivalence theorem above, m^k is D-optimal if and only if $d(\boldsymbol{\xi}, m^k) \leq n_{\mathrm{p}}$ for any $\boldsymbol{\xi} \in \Xi$. Choose $\boldsymbol{\xi}^*$ to maximize $d(\boldsymbol{\xi}, m^k)$, *i.e.*

$$\xi^* = \arg\max_{\xi\in\Xi} \frac{\partial \ln \det \mathbf{F}_{\mathrm{ps}}[\mathbf{p},\, (1-\alpha)m^k + \alpha m_\xi]}{\partial\alpha}\Big|_{\alpha=0},$$

with m_ξ the discrete measure with unit weight at ξ. This amounts to choosing the direction of steepest ascent. The structure of the algorithm is then as follows.

Step 1: Choose a discrete non-degenerate initial design measure m^1 (a normalized discrete distribution with at least n_p support points, such that $\det \mathbf{F}_{\mathrm{ps}}(\mathbf{p}, m^1) \neq 0$). Choose some positive tolerance $\delta << 1$. Set $k = 1$.
Step 2: Find $\xi^* = \arg\max_{\xi\in\Xi} d(\xi, m^k)$. If $d(\xi^*, m^k) < n_\mathrm{p} + \delta$, stop.
Step 3: Set $m^{k+1} = (1-\alpha_k)m^k + \alpha_k m_{\xi^*}$, increment k by one and go to Step 2.

The step size α_k remains to be chosen. Fedorov's algorithm (1972) uses the optimal value

$$\alpha_k^* = \arg \max_{\alpha_k\in]0,\,1[} \det \mathbf{F}_{\mathrm{ps}}(\mathbf{p}, m^{k+1}) = \frac{d(\xi^*, m^k) - n_\mathrm{p}}{n_\mathrm{p}[d(\xi^*, m^k) - 1]}.$$

Wynn's algorithm (1972) uses a predefined sequence $\{\alpha_k\}$ that satisfies

$$\alpha_k > 0, \quad \lim_{k\to\infty} \alpha_k = 0, \quad \sum_{k=1}^{\infty} \alpha_k = \infty,$$

for instance $\alpha_k = 1/(k+1)$.

The computation of $d(\xi, m^k)$ requires inversion of $\mathbf{F}_{\mathrm{ps}}(\mathbf{p}, m^k)$. Using the matrix-inversion lemma to update $\mathbf{F}_{\mathrm{ps}}^{-1}(\mathbf{p}, m^{k-1})$ to $\mathbf{F}_{\mathrm{ps}}^{-1}(\mathbf{p}, m^k)$ may be advantageous when n_p is large.

Provided that a global maximizer ξ^* is computed at Step 2, the Fedorov-Wynn algorithm converges to a D-optimal measure whatever the initial measure, a considerable improvement over the algorithms for exact design of Section 6.2.1. Note that optimization at Step 2 may be facilitated if Ξ is finite. L-optimal design measures can be determined with a similar algorithm (Fedorov, 1972), with another function $d(\xi, m)$. Algorithms for various criteria mentioned in Section 6.1 can be found in (Atwood, 1976, 1980; Pázman, 1986; Wu, 1978).

This type of algorithm never removes any support point from the design measure, which slows down convergence. Several methods have been suggested to avoid this drawback (Atwood, 1973; StJohn and Draper, 1975). Basic ideas for improvement are as follows.

If the number of support points of m^k is larger than n_p, it may be profitable to remove some of them to reduce quickly the consequences of a bad choice of the initial measure m^1, for instance with the following algorithm.

— Find

$$\xi^{i*} = \arg \min_{\xi^i\in\operatorname{supp}\{m^k\}} d(\xi^i, m^k).$$

— If $d(\xi^{i*}, m^k)$ is much smaller than n_p, remove ξ^{i*} from m^k and spread its weight μ_i^* over the most promising support points (those such that $d(\xi^i, m^k) > n_\mathrm{p}$) to get a new measure

$$m^{k+} = \left\{ \begin{matrix} \xi^i \\ \mu_i^+ \end{matrix} \quad i \neq i* \right\}$$

with weights

$$\mu_i^+ = \begin{cases} \mu_i + \mu_i^* \dfrac{d(\xi^i, m^k) - n_\mathrm{p}}{\sum\limits_{n \in \mathbb{I}} [d(\xi^n, m^k) - n_\mathrm{p}]} & \text{if } i \in \mathbb{I}, \\ \mu_i \text{ otherwise,} & \end{cases}$$

where

$$\mathbb{I} = \{ i \mid d(\xi^i, m^k) > n_\mathrm{p} \}.$$

— Compare det $\mathbf{F}_\mathrm{ps}(\mathbf{p}, m^{k+})$ with det $\mathbf{F}_\mathrm{ps}(\mathbf{p}, m^{k+1})$, where m^{k+1} is obtained from m^k by the Fedorov-Wynn algorithm. Carry forward the best measure to the next iteration.

If at iteration k some weights μ_i of the measure m^k are larger than $1/n_\mathrm{p}$, the sum of the differences $\mu_i - 1/n_\mathrm{p}$ can also be spread over the other support points.

If at Step 2 of the Fedorov-Wynn algorithm several equivalent support points ξ^* are obtained, the weight α_k (which corresponds to the step size) can be spread over them. Similarly, if the minimization of $d(\xi^i, m^k)$ yields several equivalent support points ξ^{i*}, all of them can be removed, and the sum of their weights can be spread over the remaining points.

Algorithms from another family, *exchange algorithms*, remove the weight $\alpha_k\mu_i$ at Step 3 from a single support point ξ^i of m^k. When $\alpha_k = 1$, this support point is then replaced by ξ^*. Such algorithms have been used, *e.g.*, for estimation of mixtures (see (Böhning, 1985, 1989) and Section 3.3.4), and for discrimination between model structures (Huang, 1991; Huang, Pronzato and Walter, 1991).

A particularly efficient method consists of replacing Step 3 by

Step 3': Obtain m^{k+1} by optimizing the weights of the support points $\mathrm{supp}\{m^k\} \cup \{\xi^*\}$.

This corresponds to the maximization of a concave function over a convex set (the weights are positive and their sum is equal to one), and sequential quadratic programming can be used (Section 4.3.4.5).

Algorithm for finite sets. When ξ is finite, say with cardinality N, its N points can be considered as possible support points, the weights of which have to be optimized. A constrained-optimization method (sequential quadratic programming for instance) can then be used for that purpose. It may, however, turn out to be rather inefficient if N is large.

Another approach may then be employed, close to the EM algorithm presented in Section 3.3.4. The presentation is here for D-optimality, but it can be generalized to other optimization problems with normalized distributions (Titterington, 1976; Silvey, Titterington and Torsney, 1978; Torsney, 1983, 1988; Torsney and Alahmadi, 1992).

Step 1: Choose a discrete non-degenerate initial design measure m^1, with strictly positive weight μ_i^1 at each of the N points ξ^i, choose some positive tolerance $\delta << 1$, set $k = 1$.

Step 2: If $d(\xi^i, m^k) < n_p + \delta$ for each ξ^i, stop.

Step 3: Update the weights according to

$$\mu_i^{k+1} = \mu_i^k \frac{d(\xi^i, m^k)}{n_p}, \; i = 1, \dots , N,$$

increment k by one and go to Step 2.

This algorithm converges monotonically towards a D-optimal design measure whatever m^1 (Pázman, 1986). Some weights associated with erroneous support points may nevertheless decrease very slowly, as with the Fedorov-Wynn algorithm. Convergence may then be accelerated by removing any support point whose weight is clearly tending to zero. Its weight is then spread over the remaining points. One must remember, however, that once a weight is set to zero, it will remain so for all subsequent iterations (which is why the initial weight of each ξ^i must be strictly positive).

6.3 Applications

6.3.1 Optimal measurement times

When the number of observations to be made is small, the measurement times can be individually optimized, as was done, *e.g.*, in Examples 6.2, 6.3 and 6.4. This is often the case in biology; see, *e.g.*, (Landaw, 1980). The LI character of the structure, and the simple shape of the inputs (impulse or step functions) often permit the analytical expression for the model output at any time to be derived. The sensitivity of this output and the matrix $\mathbf{F}(\mathbf{p}, \Xi)$ can then be found without difficulty.

On the other hand, engineering systems often involve experiments with a large number of observations. The optimization of measurement times will then usually be restricted to determination of an optimal sampling frequency. This problem will be considered in Section 6.3.3, simultaneously with the determination of an optimal input sequence.

6.3.2 Optimal inputs

We shall only consider systems with one input and one output, but the results easily extend to systems with several inputs and outputs. Note that, for dynamical systems, the result of each observation generally depends on previous inputs, so the experimental conditions ξ^i for observation i depend on those for others. The search will be for the optimal input from a predefined class of admissible inputs, which may be parametric (*e.g.* weighted superposition of inputs with simple shapes) or not.

6.3.2.1 Parametric inputs

The admissible inputs are defined as the superposition of basic signals with given shapes. For instance, one may consider the superposition of

- rectangular pulses $a_i[H(t-t_i) - H(t-T_i-t_i)]$, where H is the Heaviside step function, $H(t) = 0$ for $t < 0$, $H(t) = 1$ for $t \geq 0$,
- impulses $b_i\delta(t-\tau_i)$, where δ is the Dirac distribution,
- polynomials $\sum_{k=0}^{n_k} d_k t^k$.

The parameters Ξ that characterize an input u are then respectively

- the starting time t_i, amplitude a_i and duration T_i of each rectangular pulse,
- the time τ_i and area b_i of each impulse (b_i may correspond to the dose administered in biological experiments),
- the coefficients d_k of the polynomials.

The design problem then amounts to a nonlinear optimization problem with respect to Ξ. One of the main interests of this approach is the possibility of taking any kind of constraint on the input into account. The basic signals can sometimes be chosen so that the analytic expression for the associated model response is easy to derive. If the model structure is LI and the initial conditions are zero, the response to the input given by the superposition of the basic signals is the superposition of the responses to these signals. The sensitivity functions used to construct $\mathbf{F}(\mathbf{p}, \Xi)$ can then be derived analytically.

EXAMPLE 6.7

Consider the system given by

$$\frac{\mathrm{d}}{\mathrm{d}t} y_{\mathrm{m}}(t, p^*) = -p^* y_{\mathrm{m}}(t, p^*) + u(t), \quad y_{\mathrm{m}}(0_-, p^*) = 0,$$

$$y(t) = y_{\mathrm{m}}(t, p^*) + \varepsilon(t),$$

where p^* is the parameter to be estimated, with $\{\varepsilon(t)\}$ an i.i.d. sequence distributed $\mathcal{N}(0, \sigma^2)$. We wish to determine an optimal input of unit area (*i.e.* a unit dose), in the family

$$u(t) = (1-\alpha)\delta(t) + \frac{\alpha}{T}[H(t) - H(t-T)], \;\; T > 0, \; 0 \leq \alpha \leq 1.$$

Assume that only one observation is to be made. We also wish to find the optimal measurement time t, with the constraint $t \leq T$. The experiment to be performed is thus characterized by the vector $\Xi = (\alpha, T, t)^{\mathrm{T}}$. The response of the model $y_{\mathrm{m}}(t, p)$ for $0 \leq t \leq T$ is given by

$$y_{\mathrm{m}}(t, p) = \frac{\alpha}{pT} + \exp(-pt)\left[1 - \alpha\left(1 + \frac{1}{pT}\right)\right].$$

The Fisher information matrix, here a scalar, can be written as

$$\mathbf{F}(p, \Xi) = \frac{1}{\sigma^2}\left\{\exp(-pt)\left[\frac{\alpha}{p^2T} - t + \alpha t\left(1 + \frac{1}{pT}\right)\right] - \frac{\alpha}{p^2T}\right\}^2$$

and the D-optimal experiment is

$$\Xi_{\mathrm{D}} = (0, T, \frac{1}{p})^{\mathrm{T}},$$

with T arbitrary. The optimal input in the family considered is thus the unit impulse. A generalization of this property is given in (Cobelli and Thomaseth, 1985); see also (Cobelli and Thomaseth, 1988a, 1988b). ◊

In contrast to biology, for which feasible inputs generally have simple shapes, engineering may allow very large classes of admissible inputs, so the restriction to parametric inputs must be relaxed. Note, however, that determining an optimal parametric input may serve to initialize input design in a larger class. Moreover, some particular constraints on the shapes of admissible inputs can easily be taken into account for parametric inputs, but raise more difficulties for nonparametric inputs.

6.3.2.2 Nonparametric inputs

The first difficulty lies in the calculation of the Fisher information matrix $\mathbf{F}(\mathbf{p}, \Xi)$. The most general approach computes the sensitivity functions by simulating differential or difference equations (Section 4.3.3.2). The optimization of a cost function $\phi[\mathbf{F}(\mathbf{p}, \Xi)]$ may require a large number of simulations, which makes this approach rather heavy. However, an analytic expression of $\mathbf{F}(\mathbf{p}, \Xi)$ can be obtained for LI stationary models, treated in what follows. An input signal may then be characterized by its evolution as a function of time or by its spectral representation, and the two approaches will be considered in turn. See also (Mehra, 1974b; Zarrop, 1979; Titterington, 1980; Krolikowski and Eykhoff, 1985).

Time domain. We consider here discrete-time models, and the entries of the vector Ξ of experimental conditions are the input sequence $\{u(t)\}$ to be applied to the system. The dependence of $\mathbf{F}$ on Ξ is omitted to simplify notation.

EXAMPLE 6.8

Consider the FIR structure

$$y(t) = B(q, \mathbf{p}^*)u(t) + \varepsilon(t), \;\; t = 1, \ldots, n_t,$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, \sigma^2)$, with

$$B(q, \mathbf{p}) = b_1q^{-1} + \ldots + b_{n_b}q^{-n_b}.$$

The problem is to determine the input sequence to estimate $\mathbf{p} = (b_1, \ldots, b_{n_b})^{\mathrm{T}}$.

Consider first inputs with *bounded average power*

$$P_{\mathrm{u}} \leq P_{\mathrm{u}_{\max}}, \quad \text{with} \quad P_{\mathrm{u}} \approx \frac{1}{n_{\mathrm{t}}} \sum_{t=1}^{n_{\mathrm{t}}} u^2(t-\tau), \ \ \tau = 1, \dots, n_{\mathrm{b}}.$$

The model structure is LP, and the Fisher information matrix is

$$\mathbf{F}(\mathbf{p}) = \frac{1}{\sigma^2} \mathbf{R}^{\mathrm{T}}\mathbf{R},$$

with

$$\mathbf{R} = \begin{bmatrix} u(0) & u(-1) & \dots & \dots & u(1-n_{\mathrm{b}}) \\ u(1) & u(0) & \dots & \dots & u(2-n_{\mathrm{b}}) \\ \dots & \dots & \dots & \dots & \dots \\ u(n_{\mathrm{t}}-1) & \dots & \dots & \dots & u(n_{\mathrm{t}}-n_{\mathrm{b}}) \end{bmatrix},$$

which implies

$$\mathbf{F}(\mathbf{p}) = \frac{1}{\sigma^2} \sum_{t=1}^{n_{\mathrm{t}}} \begin{bmatrix} u(t-1)^2 & \dots & \dots & u(t-1)u(t-n_{\mathrm{b}}) \\ \dots & \dots & \dots & \dots \\ u(t-1)u(t-n_{\mathrm{b}}) & \dots & \dots & u(t-n_{\mathrm{b}})^2 \end{bmatrix}.$$

Since the average power of the input signal is P_{u}, the diagonal terms of $\mathbf{F}(\mathbf{p})$ are approximately equal to $P_{\mathrm{u}}n_{\mathrm{t}}/\sigma^2$. Consider the matrix $\mathbf{F}(\mathbf{p})\sigma^2/(P_{\mathrm{u}}n_{\mathrm{t}})$. Its diagonal terms are approximately equal to one, and it is positive-definite provided the input signal is rich enough. Its determinant is thus a maximum when it is equal to the identity matrix. The matrix $\mathbf{F}(\mathbf{p})$ then becomes approximately $(P_{\mathrm{u}}n_{\mathrm{t}}/\sigma^2)\mathbf{I}_{n_{\mathrm{b}}}$, the determinant of which is a maximum for $P_{\mathrm{u}} = P_{\mathrm{u}_{\max}}$. A necessary and sufficient condition for the input sequence u_{D} to be D-optimal is thus

$$\frac{1}{n_{\mathrm{t}}} \sum_{t=1}^{n_{\mathrm{t}}} u_{\mathrm{D}}(t-i)u_{\mathrm{D}}(t-k) = P_{\mathrm{u}_{\max}} \ \delta_{ik}, \ \ i, k = 1, \dots, n_{\mathrm{b}},$$

where δ_{ik} is the Kronecker delta ($\delta_{ik} = 1$ for $i = k$, 0 otherwise). When the number of observations tends to infinity, an i.i.d. sequence is thus optimal. For any finite n_{t}, a binary sequence approaching this condition can be determined (Goodwin and Payne, 1977).

Consider now the case of inputs with *bounded amplitude*,

$$-1 \leq u(t) \leq 1, \ \ \forall t\,.$$

From the discussion above, the Fisher information matrix for a D-optimal input is proportional to the average power P_{u} of the input signal, which should be maximized. This is obtained for

$$u(t) = \pm 1, \ \ \forall t\,.$$

In this case, an uncorrelated binary sequence (–1, +1) is thus D-optimal.

For both types of constraint, the optimal input does not depend on the value of the parameters b_i's since the model structure is LP. ◊

Consider now the more general case of a Box-Jenkins structure (Sections 2.4 and 3.3.2):

$$y(t) = F(q, \mathbf{p}^*)u(t) + G(q, \mathbf{p}^*)\varepsilon(t),$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, \sigma^2)$ and G is a stable rational polynomial function in q^{-1} with stable inverse. Assume that σ^2 is unknown, so that the extended parameter vector

$$\mathbf{p}_e = (\mathbf{p}^{\mathrm{T}}, \sigma^2)^{\mathrm{T}}$$

has to be estimated. Since σ^2 is unknown, the first element of the impulse response of G can be taken as one without any loss of generality.

The prediction error is

$$e_{\mathrm{p}}(t, \mathbf{p}) = G^{-1}(q, \mathbf{p})[y(t) - F(q, \mathbf{p})u(t)],$$

and forms an i.i.d. sequence when $\mathbf{p} = \mathbf{p}^*$. The log-likelihood of the n_t observations $\mathbf{y}^s$ is

$$\ln \pi(\mathbf{y}^s|\mathbf{p}_e) = -\frac{n_t}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{n_t} e_{\mathrm{p}}^2(t, \mathbf{p}).$$

The Fisher information matrix is

$$\mathbf{F}(\mathbf{p}_e) = \underset{\mathbf{y}^s|\mathbf{p}_e}{\mathrm{E}}\left\{\frac{\partial \ln \pi(\mathbf{y}^s|\mathbf{p}_e)}{\partial \mathbf{p}_e}\frac{\partial \ln \pi(\mathbf{y}^s|\mathbf{p}_e)}{\partial \mathbf{p}_e^{\mathrm{T}}}\right\},$$

with

$$\frac{\partial \ln \pi(\mathbf{y}^s|\mathbf{p}_e)}{\partial \mathbf{p}_e} = -\frac{1}{\sigma^2}\sum_{t=1}^{n_t} e_{\mathrm{p}}(t, \mathbf{p})\mathbf{s}_{e_{\mathrm{p}}}(t, \mathbf{p}_e) - \frac{n_t}{2\sigma^2}\frac{\partial \sigma^2}{\partial \mathbf{p}_e} + \frac{1}{2\sigma^4}\frac{\partial \sigma^2}{\partial \mathbf{p}_e}\sum_{t=1}^{n_t} e_{\mathrm{p}}^2(t, \mathbf{p}),$$

where

$$\mathbf{s}_{e_{\mathrm{p}}}(t, \mathbf{p}_e) = \frac{\partial e_{\mathrm{p}}(t, \mathbf{p})}{\partial \mathbf{p}_e} = \left[\frac{\partial e_{\mathrm{p}}(t, \mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}}, 0\right]^{\mathrm{T}}$$

is the sensitivity of the prediction error with respect to $\mathbf{p}_e$. Note that

$$\frac{\partial e_{\mathrm{p}}(t, \mathbf{p})}{\partial \mathbf{p}} = -G^{-1}(q, \mathbf{p})\left[\frac{\partial G(q, \mathbf{p})}{\partial \mathbf{p}} e_{\mathrm{p}}(t, \mathbf{p}) + \frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}} u(t)\right],$$

where $\partial G(q, \mathbf{p})/\partial \mathbf{p}$ and $\partial F(q, \mathbf{p})/\partial \mathbf{p}$ are vector functions of the time-delay operator q^{-1}. Since the first element of the impulse response of G is one, its derivative with respect to $\mathbf{p}$ is zero and $[\partial G(q, \mathbf{p})/\partial \mathbf{p}]e_{\mathrm{p}}(t, \mathbf{p})$ only depends on past prediction errors $e_{\mathrm{p}}(t-\tau, \mathbf{p})$ with $\tau \geq 1$. We thus obtain

$$\underset{\mathbf{y}^s|\mathbf{p}_e}{\mathrm{E}}\{e_{\mathrm{p}}(t+\tau, \mathbf{p})\mathbf{s}_{e_{\mathrm{p}}}(t, \mathbf{p}_e)\} = \mathbf{0},\ \forall\ \tau \geq 0.$$

Since the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, \sigma^2)$, we also have

$$\underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}_{\mathrm{c}}}{\mathrm{E}} \{e_{\mathrm{p}}(t+\tau, \mathbf{p})e_{\mathrm{p}}(t, \mathbf{p})\} = 0, \quad \forall\ \tau > 0,$$

$$\underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}_{\mathrm{c}}}{\mathrm{E}} \left\{e_{\mathrm{p}}^2(t, \mathbf{p})\right\} = \sigma^2, \quad \underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}_{\mathrm{c}}}{\mathrm{E}} \left\{e_{\mathrm{p}}^3(t, \mathbf{p})\right\} = 0, \quad \underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}_{\mathrm{c}}}{\mathrm{E}} \left\{e_{\mathrm{p}}^4(t, \mathbf{p})\right\} = 3\sigma^4.$$

The calculation of $\mathbf{F}(\mathbf{p}_{\mathrm{c}})$ finally yields

$$\mathbf{F}(\mathbf{p}_{\mathrm{c}}) = \underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}_{\mathrm{c}}}{\mathrm{E}} \left\{ \frac{1}{\sigma^2} \sum_{t=1}^{n_t} \mathbf{s}_{e_{\mathrm{p}}}(t, \mathbf{p}_{\mathrm{c}})\mathbf{s}_{e_{\mathrm{p}}}^{\mathrm{T}}(t, \mathbf{p}_{\mathrm{c}}) \right\} + \frac{n_t}{2\sigma^4} \left(\frac{\partial\sigma^2}{\partial\mathbf{p}_{\mathrm{c}}} \frac{\partial\sigma^2}{\partial\mathbf{p}_{\mathrm{c}}^{\mathrm{T}}}\right).$$

Provided σ^2 does not depend on $\mathbf{p}$, which is assumed in the following, this matrix can be written as

$$\mathbf{F}(\mathbf{p}_{\mathrm{c}}) = \begin{bmatrix} \mathbf{F}(\mathbf{p}) & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & \frac{n_t}{2\sigma^4} \end{bmatrix},$$

with

$$\mathbf{F}(\mathbf{p}) = \underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}}{\mathrm{E}} \left\{ \frac{1}{\sigma^2} \sum_{t=1}^{n_t} \frac{\partial e_{\mathrm{p}}(t, \mathbf{p})}{\partial \mathbf{p}} \frac{\partial e_{\mathrm{p}}(t, \mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}} \right\}.$$

The fact that σ^2 is unknown does not modify the expression for $\mathbf{F}(\mathbf{p})$. In the rest of this section we shall only consider the estimation of the vector $\mathbf{p}$ and thus the matrix $\mathbf{F}(\mathbf{p})$. In particular, the expression for $\mathbf{F}(\mathbf{p})$ will be found when the input is obtained without feedback and the rational fractions $F(q, \mathbf{p})$ and $G(q, \mathbf{p})$ have no common parameters.

REMARK 6.4

A similar result is obtained when the $\varepsilon(t)$'s are non-Gaussian i.i.d. random variables. The expression for the matrix $\mathbf{F}(\mathbf{p})$ associated with a stationary probability density function π_ε independent of $\mathbf{p}$ is simply obtained from the one above by substituting the weighting factor

$$w = \left(\int_{\mathbb{D}} \left(\frac{\mathrm{d}\pi_\varepsilon(\varepsilon)}{\mathrm{d}\varepsilon}\right)^2 \frac{1}{\pi_\varepsilon(\varepsilon)} \,\mathrm{d}\varepsilon \right)^{-1},$$

where $\mathbb{D} = \{\varepsilon \mid \pi_\varepsilon(\varepsilon) > 0\}$, for the covariance σ^2. ◊

Assume first that the input u is obtained without feedback. Then

$$\underset{\mathbf{y}^{\mathrm{s}}|\mathbf{p}}{\mathrm{E}} \{e_{\mathrm{p}}(t, \mathbf{p})u(t)\} = 0$$

and thus, taking the expression for $\partial e_{\mathrm{p}}(t, \mathbf{p})/\partial \mathbf{p}^{\mathrm{T}}$ into account, we get

$$\mathbf{F}(\mathbf{p}) = \frac{1}{\sigma^2} \sum_{t=1}^{n_t} \Big[G^{-1}(q, \mathbf{p}) \frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}} u(t)\Big]\Big[G^{-1}(q, \mathbf{p}) \frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}} u(t)\Big]^{\mathrm{T}} + \mathbf{F}_{\mathrm{c}}(\mathbf{p}),$$

where the matrix $\mathbf{F}_{\mathrm{c}}$ is independent of the input sequence u and can be written as

$$\mathbf{F}_{\mathrm{c}}(\mathbf{p}) = \operatorname*{E}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}} \Big\{\frac{1}{\sigma^2} \sum_{t=1}^{n_t} \Big[G^{-1}(q, \mathbf{p}) \frac{\partial G(q, \mathbf{p})}{\partial \mathbf{p}} e_{\mathrm{p}}(t, \mathbf{p})\Big] \times \Big[G^{-1}(q, \mathbf{p}) \frac{\partial G(q, \mathbf{p})}{\partial \mathbf{p}} e_{\mathrm{p}}(t, \mathbf{p})\Big]^{\mathrm{T}}\Big\}.$$

Assume, moreover, that $F(q, \mathbf{p})$ and $G(q, \mathbf{p})$ have no common parameter, and partition $\mathbf{p}$ into

$$\mathbf{p} = (\mathbf{p}_{\mathrm{F}}^{\mathrm{T}}, \quad \mathbf{p}_{\mathrm{G}}^{\mathrm{T}})^{\mathrm{T}},$$

where $\mathbf{p}_{\mathrm{F}}$ and $\mathbf{p}_{\mathrm{G}}$ are the parameter vectors in F and G respectively. $\mathbf{F}(\mathbf{p})$ can then be written as

$$\mathbf{F}(\mathbf{p}) = \begin{bmatrix} \mathbf{F}_{\mathrm{F}}(\mathbf{p}) & \mathbf{O} \\ \mathbf{O} & \mathbf{F}_{\mathrm{G}}(\mathbf{p}) \end{bmatrix},$$

where

$$\mathbf{F}_{\mathrm{F}}(\mathbf{p}) = \frac{1}{\sigma^2} \sum_{t=1}^{n_t} \Big[G^{-1}(q, \mathbf{p}) \frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}} u(t)\Big]\Big[G^{-1}(q, \mathbf{p}) \frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}} u(t)\Big]^{\mathrm{T}}$$

and

$$\begin{aligned} \mathbf{F}_{\mathrm{G}}(\mathbf{p}) &= \operatorname*{E}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}} \Big\{\frac{1}{\sigma^2} \sum_{t=1}^{n_t} \Big[G^{-1}(q, \mathbf{p}) \frac{\partial G(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{G}}} e_{\mathrm{p}}(t, \mathbf{p})\Big] \\ &\qquad \times \Big[G^{-1}(q, \mathbf{p}) \frac{\partial G(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{G}}} e_{\mathrm{p}}(t, \mathbf{p})\Big]^{\mathrm{T}}\Big\} \\ &= \frac{n_t}{\sigma^2} \operatorname*{E}_{\mathbf{y}^{\mathrm{s}}|\mathbf{p}} \Big\{\Big[G^{-1}(q, \mathbf{p}) \frac{\partial G(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{G}}} e_{\mathrm{p}}(t, \mathbf{p})\Big]\Big[G^{-1}(q, \mathbf{p}) \frac{\partial G(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{G}}} e_{\mathrm{p}}(t, \mathbf{p})\Big]^{\mathrm{T}}\Big\}. \end{aligned}$$

The expected value in the last term is the autocorrelation at lag zero of the prediction error filtered by $G^{-1}(q, \mathbf{p})\partial G(q, \mathbf{p})/\partial \mathbf{p}_{\mathrm{G}}$.

$\mathbf{F}_{\mathrm{G}}(\mathbf{p})$ does not depend on the choice of the input sequence (so u cannot help us estimate $\mathbf{p}_{\mathrm{G}}$ accurately), and the optimization will thus only involve a functional of the $n_{\mathrm{p_F}} \times n_{\mathrm{p_F}}$ matrix $\mathbf{F}_{\mathrm{F}}(\mathbf{p})$. The D-optimality criterion, for instance, leads to a search for the input sequence u_{D} that maximizes

$$j_{\mathrm{D}}(u) = \det \frac{1}{\sigma^2} \sum_{t=1}^{n_t} \left[G^{-1}(q, \mathbf{p}) \frac{\partial F(q, \mathbf{p})}{\mathrm{d}\mathbf{p}_{\mathrm{F}}} u(t)\right]\left[G^{-1}(q, \mathbf{p}) \frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}} u(t)\right]^{\mathrm{T}}.$$

Define

$$\mathbf{v}(t) = G^{-1}(q, \mathbf{p}) \frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}} u(t);$$

$j_{\mathrm{D}}(u)$ can then be written as

$$j_{\mathrm{D}}(u) = \det \frac{1}{\sigma^2} \sum_{t=1}^{n_t} \mathbf{v}(t)\mathbf{v}^{\mathrm{T}}(t),$$

where the components of the $\mathbf{v}(t)$'s satisfy recurrence equations, as for the sensitivity functions calculated in Section 4.3.3.2 (Goodwin, Murdoch and Payne, 1973; Goodwin and Payne, 1977). The determination of a D-optimal input sequence $\{u(t)\}$ is a nonlinear optimal control problem, which can also rely on a state-space model structure; see, *e.g.*, (Mehra, 1974a, 1974b, 1981; Kalaba and Spingarn, 1982, 1984) and the comparative study (Krolikowski and Eykhoff, 1985). Applications in biology can be found in (Kalaba and Spingarn, 1974, 1981; Cobelli and Thomaseth, 1985, 1988a, 1988b; Cobelli, Ruggeri and Thomaseth, 1984).

EXAMPLE 6.9

Consider the ARX structure

$$A(q, \mathbf{p}_{\mathrm{A}}^{*})y(t) = B(q, \mathbf{p}_{\mathrm{B}}^{*})u(t) + \varepsilon(t),$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, \sigma^2)$. Let

$$\mathbf{p} = (a_1, a_2, \ldots, a_{n_a}, b_1, b_2, \ldots, b_{n_b})^{\mathrm{T}}$$

and

$$\mathbf{p}_{\mathrm{c}} = (\mathbf{p}^{\mathrm{T}}, \sigma^2)^{\mathrm{T}}.$$

One easily obtains

$$\frac{\partial e_{\mathrm{p}}(t, \mathbf{p})}{\partial \mathbf{p}} = \begin{bmatrix} \dfrac{B(q, \mathbf{p}_{\mathrm{B}})}{A(q, \mathbf{p}_{\mathrm{A}})} u(t-1) + \dfrac{e_{\mathrm{p}}(t-1, \mathbf{p})}{A(q, \mathbf{p}_{\mathrm{A}})} \\ \cdots \\ \dfrac{B(q, \mathbf{p}_{\mathrm{B}})}{A(q, \mathbf{p}_{\mathrm{A}})} u(t-n_a) + \dfrac{e_{\mathrm{p}}(t-n_a, \mathbf{p})}{A(q, \mathbf{p}_{\mathrm{A}})} \\ -u(t-1) \\ \cdots \\ -u(t-n_b) \end{bmatrix},$$

which yields

$$\mathbf{F}(\mathbf{p}_c) = \begin{bmatrix} \mathbf{F}(\mathbf{p}) & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & \dfrac{n_t}{2\sigma^4} \end{bmatrix},$$

with

$$\mathbf{F}(\mathbf{p}) = \frac{1}{\sigma^2}\sum_{t=1}^{n_t} \mathbf{v}_F(t)\mathbf{v}_F^{\mathrm{T}}(t) + \mathop{\mathrm{E}}_{\mathbf{y}^s|\mathbf{p}} \left\{\frac{1}{\sigma^2}\sum_{t=1}^{n_t} \mathbf{v}_G(t)\mathbf{v}_G^{\mathrm{T}}(t)\right\},$$

where

$$\mathbf{v}_F(t) = \begin{bmatrix} \dfrac{B(q, \mathbf{p}_B)}{A(q, \mathbf{p}_A)} u(t-1) \\ \cdots \\ \dfrac{B(q, \mathbf{p}_B)}{A(q, \mathbf{p}_A)} u(t-n_a) \\ -u(t-1) \\ \cdots \\ -u(t-n_b) \end{bmatrix} \quad \text{and} \quad \mathbf{v}_G(t) = \begin{bmatrix} \dfrac{e_p(t-1, \mathbf{p})}{A(q, \mathbf{p}_A)} \\ \cdots \\ \dfrac{e_p(t-n_a, \mathbf{p})}{A(q, \mathbf{p}_A)} \\ 0 \\ \cdots \\ 0 \end{bmatrix}.$$

Since this is a special case of the Box-Jenkins model structure, with $F(q, \mathbf{p}) = B(q, \mathbf{p}_B)/A(q, \mathbf{p}_A)$ and $G(q, \mathbf{p}) = 1/A(q, \mathbf{p}_A)$, the matrix $\mathbf{F}(\mathbf{p})$ can also be written as

$$\mathbf{F}(\mathbf{p}) = \frac{1}{\sigma^2}\sum_{t=1}^{n_t} \left[G^{-1}(q, \mathbf{p})\frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}} u(t)\right]\left[G^{-1}(q, \mathbf{p})\frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}} u(t)\right]^{\mathrm{T}} + \mathbf{F}_G(\mathbf{p}),$$

with

$$\mathbf{F}_G(\mathbf{p}) = \mathop{\mathrm{E}}_{\mathbf{y}^s|\mathbf{p}} \left\{\frac{1}{\sigma^2}\sum_{t=1}^{n_t} \left[G^{-1}(q, \mathbf{p})\frac{\partial G(q, \mathbf{p})}{\partial \mathbf{p}} e_p(t, \mathbf{p})\right] \times \left[G^{-1}(q, \mathbf{p})\frac{\partial G(q, \mathbf{p})}{\partial \mathbf{p}} e_p(t, \mathbf{p})\right]^{\mathrm{T}}\right\}.$$

However, F and G now have common parameters, so $\mathbf{F}_G(\mathbf{p})$ affects the choice of the optimal input sequence.

In the rest of the example, we consider the particular case $n_a = n_b = 1$, *i.e.*

$$y(t+1) = -ay(t) + bu(t) + \varepsilon(t+1).$$

We wish to find a sequence $u(1), \ldots, u(n_t)$ that maximizes det $\mathbf{F}(\mathbf{p})$, with $\mathbf{p} = (a, b)^{\mathrm{T}}$. Inputs have either bounded amplitude or bounded average power. The matrix $\mathbf{F}(\mathbf{p})$ can be written as

$$\mathbf{F}(\mathbf{p}) = \frac{1}{\sigma^2}\sum_{t=2}^{n_t+1} \begin{bmatrix} y_m^2(t-1) + \sigma^2\dfrac{1 - a^{2t}}{1 - a^2} & -u(t-1)y_m(t-1) \\ -u(t-1)y_m(t-1) & u^2(t-1) \end{bmatrix},$$

with $A(q, a)y_{\mathrm{m}}(t) = B(q, b)u(t)$, *i.e.*

$$y_{\mathrm{m}}(t-1) = -ay_{\mathrm{m}}(t-2) + bu(t-2) \text{ for } t > 2, \quad \text{and} \quad y_{\mathrm{m}}(t-1) = 0 \text{ for } t \leq 2.$$

Let $\mathbf{F}(\mathbf{p}, k)$ be the Fisher information matrix associated with the first k data points (so $\mathbf{F}(\mathbf{p}, n_{\mathrm{t}}) = \mathbf{F}(\mathbf{p})$), and define the state vector

$$\mathbf{x}(k) = [y_{\mathrm{m}}(k), F_{11}(\mathbf{p}, k), F_{12}(\mathbf{p}, k), F_{22}(\mathbf{p}, k)]^{\mathrm{T}}.$$

It satisfies the nonlinear recurrence equation

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), u(k)] = \begin{bmatrix} -ax_1(k) + bu(k) \\ x_2(k) + \dfrac{1}{\sigma^2}x_1^2(k) + \dfrac{1 - a^{2(k+1)}}{1 - a^2} \\ x_3(k) - \dfrac{1}{\sigma^2}u(k)x_1(k) \\ x_4(k) + \dfrac{1}{\sigma^2}u^2(k) \end{bmatrix},$$

with $\mathbf{x}(k) = \mathbf{0}$ for $k \leq 1$. The determinant of the Fisher information matrix $\mathbf{F}(\mathbf{p})$ is

$$\det \mathbf{F}(\mathbf{p}, n_{\mathrm{t}}) = x_2(n_{\mathrm{t}}+1)x_4(n_{\mathrm{t}}+1) - x_3^2(n_{\mathrm{t}}+1).$$

The Lagrangian to be minimized with respect to the input sequence is

$$\mathcal{L} = -[x_2(n_{\mathrm{t}}+1)x_4(n_{\mathrm{t}}+1) - x_3^2(n_{\mathrm{t}}+1)] + \sum_{k=1}^{n_{\mathrm{t}}} \boldsymbol{\lambda}^{\mathrm{T}}(k)\{\mathbf{f}[\mathbf{x}(k), u(k)] - \mathbf{x}(k+1)\},$$

with $\boldsymbol{\lambda}(k)$ the adjoint state vector.

Consider first the *amplitude constraints*,

$$-1 \leq u(k) \leq 1, \quad k = 1, \ldots, n_{\mathrm{t}}.$$

The gradient of the Lagrangian with respect to the input is

$$\frac{\partial \mathcal{L}}{\partial u(k)} = \boldsymbol{\lambda}^{\mathrm{T}}(k)\frac{\partial \mathbf{f}}{\partial u}\Big|_{\mathbf{x}(k),\, u(k)}, \quad k = 1, \ldots, n_{\mathrm{t}},$$

and the adjoint state is such that

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}(k)} = \mathbf{0}, \quad k = 2, \ldots, n_{\mathrm{t}}.$$

It thus satisfies the backward-in-time equation

$$\boldsymbol{\lambda}(k-1) = \frac{\partial \mathbf{f}^{\mathrm{T}}}{\partial \mathbf{x}}\Big|_{\mathbf{x}(k),\, u(k)} \boldsymbol{\lambda}(k),$$

or equivalently

$$\boldsymbol{\lambda}(k-1) = \begin{bmatrix} -a & \frac{2x_1(k)}{\sigma^2} & -\frac{u(t)}{\sigma^2} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{\lambda}(k), \quad k = 2, \dots, n_t,$$

with the terminal condition $\partial \mathcal{L}/\partial \mathbf{x}(n_t+1) = \mathbf{0}$, *i.e.*

$$\boldsymbol{\lambda}(n_t) = [0, -x_4(n_t+1), 2x_3(n_t+1), -x_2(n_t+1)]^{\mathrm{T}}.$$

We thus obtain for $k = 1, \dots, n_t$

$$\lambda_2(k) = \lambda_2 = -x_4(n_t+1),\ \lambda_3(k) = \lambda_3 = 2x_3(n_t+1),\ \lambda_4(k) = \lambda_4 = -x_2(n_t+1),$$

and

$$\lambda_1(k-1) = -a\lambda_1(k) - \frac{2x_4(n_t+1)x_1(k)}{\sigma^2} - \frac{2x_3(n_t+1)u(k)}{\sigma^2}, \quad k = 2, \dots, n_t,$$

with $\lambda_1(n_t) = 0$. The gradient of the Lagrangian with respect to the input is finally given by

$$\frac{\partial \mathcal{L}}{\partial u(k)} = b\lambda_1(k) - \frac{2x_3(n_t+1)x_1(k)}{\sigma^2} - \frac{2x_2(n_t+1)u(k)}{\sigma^2}, \quad k = 1, \dots, n_t,$$

which can be computed for any sequence of inputs by simulation of the forward-in-time system $\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), u(k)]$, followed by simulation of the backward-in-time recurrence equation for $\lambda_1(k)$. This corresponds to the adjoint-state method for the calculation of gradients, presented in a more general context in Section 4.3.3.2. The adjoint-code method of the same section could be used as well, with the direct code computing det $\mathbf{F}(\mathbf{p}, n_t)$ by simulation of the recurrence equation $\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), u(k)]$.

An optimal input sequence can be determined by, *e.g.*, a gradient-projection algorithm (Section 4.3.4.4), with easy implementation here due to the simple form of the admissible domain for the input. Goodwin, Murdoch and Payne (1973) suggest the simple rule

$$u(k) = -\operatorname{sign}\left[\frac{\partial \mathcal{L}}{\partial u(k)}\right].$$

Note that the cost function is not convex (which is common for exact design problems), and the solution may turn out to be only locally optimal. Figure 6.5 presents the cost contours for det $\mathbf{F}(\mathbf{p}, 3)$ in the plane defined by $u(2)$ and $u(3)$ when $u(1)$ is set equal to one ($a = 0.1$, $b = 1$, $\sigma^2 = 0.01$). The sequence (1, 1, −1) is locally optimal; the global optimum corresponds to (1, −1, −1). Optimization should therefore be repeated with various initial points.

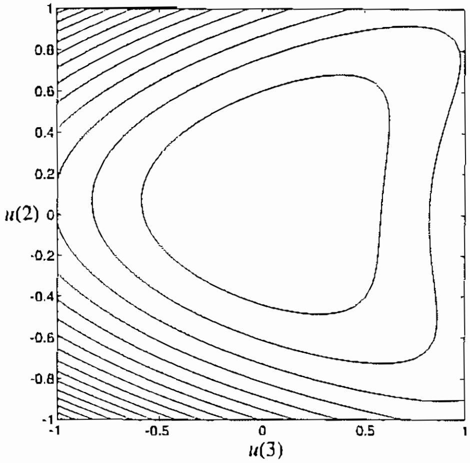


Figure 6.5. Non-convexity of the design problem ($u(1) = 1$)

Consider now a constraint on the *average power* of the input signal,

$$\frac{1}{n_t}\sum_{k=1}^{n_t} u^2(k) \leq P_{\mathrm{u}_{\max}}.$$

A first possibility would be to use a constrained optimization method for the whole input sequence, with the gradient of the cost computed as previously. However, the dimension of the space can be reduced by taking advantage of the boundary conditions.

With the same notation as previously, $x_4(n_t+1) = n_t P_{\mathrm{u}_{\max}}/\sigma^2$, and the terminal condition on λ_4 is thus removed. One obtains

$$\lambda_2(k) = \lambda_2 = -\frac{n_t P_{\mathrm{u}_{\max}}}{\sigma^2},\ \lambda_3(k) = \lambda_3 = 2x_3(n_t+1),\ \lambda_4(k) = \lambda_4,\quad k = 1, \ldots, n_t,$$

with λ_4 free, and

$$\lambda_1(k-1) = -a\lambda_1(k) - \frac{2n_t P_{\mathrm{u}_{\max}} x_1(k)}{\sigma^4} - \frac{\lambda_3 u(k)}{\sigma^2},\quad k = 2, \ldots, n_t,$$

with $\lambda_1(n_t) = 0$. The gradient of the Lagrangian with respect to the input is now given by

$$\frac{\partial \mathcal{L}}{\partial u(k)} = b\lambda_1(k) - \frac{\lambda_3 x_1(k)}{\sigma^2} + \frac{2\lambda_4 u(k)}{\sigma^2},\quad k = 1, \ldots, n_t.$$

From $\partial \mathcal{L}/\partial u(k) = 0$, one obtains

$$u(k) = \frac{\sigma^2}{2\lambda_4}\left[\frac{\lambda_3 x_1(k)}{\sigma^2} - b\lambda_1(k)\right] = h[x_1(k),\, \lambda_1(k)].$$

Substituting this expression for $u(k)$ into the backward-in-time equation for $\lambda_1(k)$, and inverting the result gives the forward-in-time equation

$$\lambda_1(k) = \left(\frac{b\lambda_3}{2\lambda_4} - a\right)^{-1}\left[\lambda_1(k-1) + 2\Big(\frac{n_t P_{\mathrm{u}_{\max}}}{\sigma^2} + \frac{\lambda_3^2}{4\lambda_4}\Big)x_1(k)\right] = g[x_1(k),\, \lambda_1(k-1)].$$

The problem is now to find initial conditions $\lambda_1(1)$, λ_3 and λ_4 such that the terminal conditions $\lambda_1(n_t) = 0$, $x_3(n_t+1) = \lambda_3/2$ and $x_4(n_t+1) = n_t P_{\mathrm{u}_{\max}}/\sigma^2$ are satisfied by the forward-in-time system

$$\left\{\begin{aligned} u(k) &= h[x_1(k),\, \lambda_1(k)], \\ \mathbf{x}(k+1) &= \mathbf{f}[\mathbf{x}(k),\, u(k)], \\ \lambda_1(k+1) &= g[x_1(k+1),\, \lambda_1(k)]. \end{aligned}\right.$$

With $\boldsymbol{\omega} = [\lambda_1(1),\, \lambda_3,\, \lambda_4]^{\mathrm{T}}$, the condition to be satisfied can be written as $\boldsymbol{\Psi}(\boldsymbol{\omega}) = \mathbf{0}$, where

$$\boldsymbol{\Psi}(\boldsymbol{\omega}) = \left[\lambda_1(n_t),\, x_3(n_t+1) - \frac{\lambda_3}{2},\, x_4(n_t+1) - \frac{n_t P_{\mathrm{u}_{\max}}}{\sigma^2}\right]^{\mathrm{T}}.$$

It can be solved for instance by the Newton-Raphson method; see, *e.g.*, (Press *et al.*, 1986). The Jacobian matrix $\partial\boldsymbol{\Psi}/\partial\boldsymbol{\omega}^{\mathrm{T}}$ must then be computed. Approximation by finite differences is often used. However, since $\boldsymbol{\Psi}$ is computed by a code that simulates the nonlinear system above, the adjoint-code technique (Section 4.3.3.2) permits computation of $\partial\boldsymbol{\Psi}/\partial\boldsymbol{\omega}^{\mathrm{T}}$ without any approximation. Note that again the solution obtained by the Newton-Raphson method may prove only locally optimal. ◊

The complexity of this approach can sometimes be avoided when the control law of the system can be determined and implemented on-line. A recursive characterization of the uncertainty in the parameters then allows, as we shall see, computation at each time instant t of the input $u(t)$ to be applied (Keviczky, 1975). In general, however, this one-step-ahead policy is not globally optimal.

Sequential design: on-line control. Consider again the ARX structure

$$A(q,\, \mathbf{p}_{\mathrm{A}}^*)y(t) = B(q,\, \mathbf{p}_{\mathrm{B}}^*)u(t) + \varepsilon(t),$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0,\, \sigma^2)$. This recurrence equation can be written as

$$y(t+1) = y_{\mathrm{m}}(t+1, \mathbf{p}^*) + \varepsilon(t+1),$$

with

$$y_{\mathrm{m}}(t+1, \mathbf{p}) = \mathbf{r}^{\mathrm{T}}(t)\mathbf{p},$$

$$\mathbf{r}(t) = [u(t), u(t-1), \ldots , u(t+1-n_{\mathrm{b}}), -y(t), -y(t-1), \ldots , -y(t+1-n_{\mathrm{a}})]^{\mathrm{T}}$$

and

$$\mathbf{p} = (b_1, b_2, \ldots , b_{n_{\mathrm{b}}}, a_1, a_2, \ldots , a_{n_{\mathrm{a}}})^{\mathrm{T}}.$$

The recursive least-squares algorithm may be used to estimate the parameters of this LP structure (Section 4.1.4). In particular, up to multiplication by σ^2, the covariance matrix of the parameter estimates satisfies

$$\mathbf{P}(t+1) = \mathbf{P}(t) - \frac{\mathbf{P}(t)\mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)}{1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)}.$$

The problem considered here is to choose at each time instant t the input $u(t)$ that satisfies the constraints and minimizes a scalar function of $\mathbf{P}(t+1)$. Consider D-optimal design with amplitude constraints

$$-1 \leq u(t) \leq 1.$$

The cost is

$$\det \mathbf{P}(t+1) = \det \mathbf{P}(t) \det\left[\mathbf{I}_{n_{\mathrm{a}}+n_{\mathrm{b}}} - \frac{\mathbf{P}(t)\mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t)}{1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)}\right],$$

or equivalently, since $\det(\mathbf{I} + \mathbf{v}_1\mathbf{v}_2^{\mathrm{T}}) = 1 + \mathbf{v}_2^{\mathrm{T}}\mathbf{v}_1$,

$$\det \mathbf{P}(t+1) = \frac{1}{1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)} \det \mathbf{P}(t).$$

The (one-step-ahead) D-optimal input $u_{\mathrm{D}}(t)$ should thus maximize

$$\mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t) = p_{11}(t)u^2(t) + 2\mathbf{p}_{21}^{\mathrm{T}}(t)\mathbf{r}_{-1}(t)u(t) + \mathbf{r}_{-1}^{\mathrm{T}}(t)\mathbf{P}_{22}(t)\mathbf{r}_{-1}(t),$$

where

$$\mathbf{r}_{-1}(t) = [u(t-1), \ldots , u(t+1-\mathrm{n_b}), -y(t), -y(t-1), \ldots , -y(t+1-n_{\mathrm{a}})]^{\mathrm{T}},$$

and

$$\mathbf{P}(t) = \begin{bmatrix} p_{11}(t) & \mathbf{p}_{21}^{\mathrm{T}}(t) \\ \mathbf{p}_{21}(t) & \mathbf{P}_{22}(t) \end{bmatrix}.$$

The scalar function $\mathbf{r}^{\mathrm{T}}(t)\mathbf{P}(t)\mathbf{r}(t)$ is quadratic in $u(t)$, and has its minimum at

$$u^*(t) = -\frac{\mathbf{p}_{21}^{\mathrm{T}}(t)\mathbf{r}_{-1}(t)}{p_{11}(t)}.$$

The D-optimal solution is thus

$$u_{\mathrm{D}}(t) = \begin{cases} +1 \text{ if } u^*(t) < 0, \\ -1 \text{ if } u^*(t) > 0, \end{cases}$$

and $u_{\mathrm{D}}(t) = \pm 1$ if $u^*(t) = 0$.

REMARKS 6.5

- — The same sequential approach may be employed with the OBE or EPC presented in Section 5.4.1.1 for parameter bounding (Pronzato and Walter, 1991b).
- — The characterization of uncertainty is exact, and much simpler than that in Example 6.9, which also concerned an ARX structure. The reason is that we use here the *observed* information matrix $\sigma^{-2}\mathbf{P}^{-1}$, which is determined on-line and depends on past observations, whereas in Example 6.9 the uncertainty in $\mathbf{p}$ had to be evaluated before making any observation, through the *expected* information matrix $\mathbf{F}(\mathbf{p})$.
- — The approach above is myopic (only one-step-ahead optimal), and better performance may be obtained with an uncorrelated binary sequence (–1, +1).
- — When the model structure is not LP, the estimation is generally not recursive. However, when the estimation is by the recursive techniques of Section 4.3.8 and parameter uncertainty is characterized by the Fisher information matrix, sequential design can be used as for an LP structure.
- — The sequential design of a globally optimal input sequence is generally extremely difficult, as it is a stochastic dynamic programming problem. Only very simple examples have been treated; see (Zacks, 1977; Pronzato, Walter and Kulcsár, 1993; Kulcsár, Pronzato and Walter, 1994) for examples with $n_t = 2$. Suboptimal solutions taking into account more than one step ahead can be determined, *e.g.*, through approximation of the posterior density of the parameters (Kulcsár, Pronzato and Walter, 1995; Pronzato, Kulcsár, and Walter, 1996). ◊

When no recursive characterization of the uncertainty in the parameters is available, Goodwin and Payne (1977) suggest recursively constructing a bounded signal $u(t)$ (*e.g.*, a binary sequence) such that its power spectrum tends to that of the optimal input signal as n_t tends to infinity. We shall now investigate the characterization of the optimal input sequence through its power spectrum.

Frequency domain. Consider again the matrix $\mathbf{F}_{\mathrm{F}}(\mathbf{p})$ obtained for the general Box-Jenkins structure without feedback

$$\mathbf{F}_{\mathrm{F}}(\mathbf{p}) = \frac{1}{\sigma^2} \sum_{t=1}^{n_t} \left[G^{-1}(q,\mathbf{p}) \frac{\partial F(q,\mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}} u(t)\right]\left[G^{-1}(q,\mathbf{p}) \frac{\partial F(q,\mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}} u(t)\right]^{\mathrm{T}}.$$

Assume that sampling is uniform, with period T, and let the number of observations tend to infinity. The matrix

$$\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p}) = \lim_{n_t \to \infty} \frac{1}{Tn_t} \mathbf{F}_{\mathrm{F}}(\mathbf{p})$$

is thus the average Fisher information matrix per unit time. If

$$\mathbf{v}(t) = G^{-1}(q, \mathbf{p})\frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}}\, u(t),$$

the term

$$\lim_{n_t \to \infty} \frac{1}{n_t}\sum_{t=1}^{n_t} \left[G^{-1}(q, \mathbf{p})\frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}}\, u(t)\right]\left[G^{-1}(q, \mathbf{p})\frac{\partial F(q, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}}\, u(t)\right]^{\mathrm{T}}$$

corresponds to the correlation matrix of the signal $\mathbf{v}$ at lag zero. From Parseval's theorem, it is equal to the integral of the power spectral density $\mathcal{P}_{\mathrm{v}}(\omega)$ of this signal.

Introduce the normalized angular frequency $\overline{\omega} = T\omega$. The matrix $\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p})$ can then be written as

$$\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p}) = \frac{1}{2\pi T\sigma^2}\int_{-\pi/T}^{\pi/T} \mathcal{P}_{\mathrm{v}}(\omega)\mathrm{d}\omega = \frac{1}{2\pi\sigma^2}\int_{-\pi}^{\pi} \mathcal{P}_{\mathrm{v}}(\overline{\omega})\mathrm{d}\overline{\omega}$$

$$= \frac{1}{2\pi\sigma^2}\int_{-\pi}^{\pi} \frac{\partial F(\mathrm{e}^{j\overline{\omega}}, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}}\, G^{-1}(\mathrm{e}^{j\overline{\omega}}, \mathbf{p})G^{-1}(\mathrm{e}^{-j\overline{\omega}}, \mathbf{p})\, \frac{\partial F(\mathrm{e}^{-j\overline{\omega}}, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}^{\mathrm{T}}}\, p_{\mathrm{u}}(\overline{\omega})\mathrm{d}\overline{\omega}),$$

where $p_{\mathrm{u}}(\overline{\omega})$ is the power spectral density of the input signal u. In what follows, the normalized angular frequency will be denoted simply by ω. Since u is real, $p_{\mathrm{u}}(\omega)$ is real and even, and $\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p})$ becomes

$$\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p}) = \frac{1}{\pi}\int_{0}^{\pi} \tilde{\mathbf{F}}_{\mathrm{F}}(\mathbf{p}, \omega)p_{\mathrm{u}}(\omega)\mathrm{d}\omega,$$

with

$$\tilde{\mathbf{F}}_{\mathrm{F}}(\mathbf{p}, \omega) = \frac{1}{\sigma^2}\,\mathrm{Re}\left[\frac{\partial \mathrm{F}(\mathrm{e}^{j\omega}, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}}\, G^{-1}(\mathrm{e}^{j\omega}, \mathbf{p})G^{-1}(\mathrm{e}^{-j\omega}, \mathbf{p})\frac{\partial F(\mathrm{e}^{-j\omega}, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}^{\mathrm{T}}}\right],$$

where Re(x) is the real part of x.

Consider now the case of inputs with unit average power, *i.e.*

$$P_{\mathrm{u}} = \frac{1}{\pi}\int_{0}^{\pi} p_{\mathrm{u}}(\omega)\mathrm{d}\omega = 1.$$

One can then show (Goodwin and Payne, 1977; Zarrop, 1979) that the set of matrices $\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p})$ is the convex hull of the set of matrices obtained for sinusoidal inputs (each of which corresponds to a spectral line). From Caratheodory's theorem, any matrix $\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p})$ may thus be written as the sum of at most $[n_{\mathrm{p_F}}(n_{\mathrm{p_F}} + 1)/2] + 1$ matrices, each obtained for a sinusoidal input. This holds true in particular for the matrix associated with the D-optimal input signal. Taking the special structure of $\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p})$ into account (F and G are assumed to have no common parameters) the number of sinusoids required can still be reduced. Indeed, one can show that any matrix $\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p})$ belongs to an affine manifold with dimension $n_{\mathrm{p_F}}$, so $n_{\mathrm{p_F}}$ distinct frequencies are enough to construct an optimal input signal (Goodwin and Payne, 1977; Zarrop, 1979). Any matrix $\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p})$ can thus be written as

$$\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p}) = \frac{1}{\sigma^2}\sum_{i=1}^{n_\omega} \mu_i \operatorname{Re}\left[\mathbf{v}(j\omega_i)\mathbf{v}^{\mathrm{T}}(-j\omega_i)\right], \quad n_\omega \leq n_{\mathrm{p_F}},$$

with $\mathbf{v}(j\omega)$ given by

$$\mathbf{v}(j\omega) = G^{-1}(\mathrm{e}^{j\omega}, \mathbf{p})\frac{\partial F(\mathrm{e}^{j\omega}, \mathbf{p})}{\partial \mathbf{p}_{\mathrm{F}}}.$$

The search for an optimal input can thus be performed in a $2n_{\mathrm{p_F}}$-dimensional space (the frequencies $\omega_i/2\pi$ and average powers μ_i of the $n_{\mathrm{p_F}}$ sinusoids). This corresponds to the continuous-design approach presented in Section 6.2.2.1. The distribution $m(\mathrm{d}\xi)$ of experimental effort is replaced by $p_{\mathrm{u}}(\omega)\mathrm{d}\omega$, with $p_{\mathrm{u}}(\omega)$ the power spectral density of the signal $u(t)$. The optimal spectrum is discrete and characterized by

$$p_{\mathrm{uD}} = \left\{\begin{bmatrix}\omega_i\\ \mu_i\end{bmatrix}, \; i = 1, \; \ldots \;, \; n_\omega\right\},$$

which is a condensed notation for

$$p_{\mathrm{uD}}(\omega) = \pi\sum_{i=1}^{n_\omega} \mu_i \left[\delta(\omega + \omega_i + \delta(\omega - \omega_i)\right], \text{ with } \sum_{i=1}^{n_\omega} \mu_i = 1,$$

where δ is the Dirac delta. Although this optimal experiment corresponds to approximate design theory (Section 6.2.2.2), it can be performed without approximation if the observations are continuous and the system is stable enough for the initial conditions to have no influence. The optimal input can be implemented in either of two ways:

— by application of a combination of n_ω sinusoids characterized by $\{\omega_i, \mu_i\}$,

— by successive application of n_ω sinusoids, with different frequencies $\omega_i/2\pi$ but the same power, each for a duration proportional to μ_i.

The similarity to approximate design permits use of the optimization algorithms presented in Section 6.2.2.4 to determine optimal input spectra (Mehra 1974b, 1981; Zarrop, 1979). As already mentioned, these algorithms yield globally optimal design measures.

REMARK 6.6

The average power of a sinusoid with amplitude λ and frequency $\omega_{\mathrm{u}}/2\pi$ is $\lambda^2/2$ if $\omega_{\mathrm{u}} \neq 0$ and λ^2 if $\omega_{\mathrm{u}} = 0$. Its power spectral density is

$$p_u(\omega) = \begin{cases} \dfrac{\pi\lambda^2}{2}[\delta(\omega+\omega_u)+\delta(\omega-\omega_u)] \text{ if } \omega_u \neq 0 \\ 2\pi\lambda^2\delta(\omega) \text{ otherwise.} \end{cases} \qquad \lozenge$$

EXAMPLE 6.9 (continued)

Here the matrix $\overline{\mathbf{F}}_F(\mathbf{p})$ is

$$\overline{\mathbf{F}}_F(\mathbf{p}) = \frac{1}{\pi\sigma^2}\int_0^{\pi}\begin{bmatrix} \dfrac{b^2}{1+a^2+2\cos(\omega)} + \dfrac{\sigma^2}{1-a^2} & -b\,\dfrac{a+\cos(\omega)}{1+a^2+2\cos(\omega)} \\ -b\,\dfrac{a+\cos(\omega)}{1+a^2+2\cos(\omega)} & 1 \end{bmatrix} p_u(\omega)\mathrm{d}\omega.$$

For a single sinusoid (which may suffice since only two parameters are to be estimated, see Remark 6.7 below) the D-optimal angular frequency is

$$\omega_D = \text{arc cos}\left[-\frac{2a}{1+a^2}\right]. \qquad \lozenge$$

EXAMPLE 6.10

Consider the same model structure as in Example 6.8, with $n_b = 3$. For the general notation for a Box-Jenkins structure, this corresponds to

$$F(q,\mathbf{p}) = B(q,\mathbf{p}) = b_1q^{-1} + b_2q^{-2} + b_3q^{-3},\ \ G(q,\mathbf{p}) = 1,$$

$$\mathbf{p} = \mathbf{p}_F = (b_1, b_2, b_3)^T.$$

The matrix $\tilde{\mathbf{F}}_F(\mathbf{p},\omega)$ corresponding to a sinusoid with angular frequency ω can be written as

$$\tilde{\mathbf{F}}_F(\mathbf{p},\omega) = \frac{1}{\sigma^2}\text{Re}\{[\exp(-j\omega), \exp(-2j\omega), \exp(-3j\omega)]^T \times [\exp(j\omega), \exp(2j\omega), \exp(3j\omega)]\}$$

or equivalently

$$\tilde{\mathbf{F}}_F(\mathbf{p},\omega) = \frac{1}{\sigma^2}\begin{bmatrix} 1 & \cos(\omega) & \cos(2\omega) \\ \cos(\omega) & 1 & \cos(\omega) \\ \cos(2\omega) & \cos(\omega) & 1 \end{bmatrix}.$$

When sampling at frequency $1/T$, the matrix $\overline{\mathbf{F}}_F(\mathbf{p})$ thus takes the form

$$\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p}) = \frac{1}{\sigma^2}\begin{bmatrix} 1 & x_1 & x_2 \\ x_1 & 1 & x_1 \\ x_2 & x_1 & 1 \end{bmatrix},$$

with

$$x_1 = \frac{1}{\pi}\int_0^{\pi} \cos(\omega)p_{\mathrm{u}}(\omega)\mathrm{d}\omega \quad \text{and} \quad x_2 = \frac{1}{\pi}\int_0^{\pi} \cos(2\omega)p_{\mathrm{u}}(\omega)\mathrm{d}\omega.$$

The maximum of det $\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p})$ is reached when $\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p})$ is proportional to the identity matrix, *i.e.* when $x_1 = x_2 = 0$ (Goodwin and Payne, 1977). This corresponds to $p_{\mathrm{u}}(\omega)$ constant on $[0, \pi]$ (white noise), and confirms the result of Example 6.8. The D-optimal input, however, is not unique. Identical performance is achieved for the input defined by the sum of two sinusoids, at normalized angular frequencies $\pi/4$ and $3\pi/4$:

$$u(t) = \cos(\frac{\pi}{4T}t) + \cos(\frac{3\pi}{4T}t),$$

or equivalently

$$u(t) = \begin{cases} 2 \text{ if } t = 0 \text{ [modulo } 8T], \\ -2 \text{ if } t = 4T \text{ [modulo } 8T], \\ 0 \text{ otherwise.} \end{cases}$$ ◊

REMARK 6.7

A condition for *persistency of excitation* can be derived from the expression

$$\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p}) = \frac{1}{\sigma^2}\sum_{i=1}^{n_\omega} \mu_i \,\mathrm{Re}[\mathbf{v}(j\omega_i)\mathbf{v}^{\mathrm{T}}(-j\omega_i)]$$

for the Fisher information matrix (Goodwin and Payne, 1977). Indeed, each term $\mathrm{Re}[\mathbf{v}(j\omega_i)\mathbf{v}^{\mathrm{T}}(-j\omega_i)]$ has rank two if $\omega_i \neq 0$ and rank one if $\omega_i = 0$. $\overline{\mathbf{F}}_{\mathrm{F}}(\mathbf{p})$ is thus singular if and only if the number n_ω of distinct frequencies in the input signal is such that $2n_\omega < n_{\mathrm{pF}}$ (the frequencies 0 or 1/2, *i.e.* $\omega = 0$ or π, count for 1/2). ◊

The results above extend straightforwardly to continuous-time systems by replacing

— $q = \exp(j\omega)$ by $s = j\omega$ in all expressions involving frequency,
— π by ∞ in the bounds of integrals.

EXAMPLE 6.11

Consider the LI system defined by

$$\tau^*\frac{\mathrm{d}}{\mathrm{d}t}y_{\mathrm{m}}(t, \tau^*) + y_{\mathrm{m}}(t, \tau^*) = u(t), \;\; y_{\mathrm{m}}(0, \tau^*) = 0,$$

$$y(t) = y_m(t, \tau^*) + \eta(t),$$

where the noise $\eta(t)$ has a power spectral density $1/(1 + a^2\omega^2)$. The transfer function of the model is given by

$$F(j\omega, \tau) = \frac{1}{1 + j\tau\omega} .$$

We wish to choose u, with unit average power, so as to estimate τ as well as possible. From the results above, since $n_{\text{pF}} = 1$, one sinusoid is enough. For a sinusoid with frequency $\omega/2\pi$,

$$\overline{\mathbf{F}}_{\text{F}}(\mathbf{p}) = \frac{-j\omega}{(1 + j\omega\tau)^2}(1 + j\omega a)(1 - j\omega a)\frac{j\omega}{(1 - j\omega\tau)^2} = \frac{\omega^2(1 + a^2\omega^2)}{(1 + \tau^2\omega^2)^2}.$$

Maximization of the (scalar) $\overline{\mathbf{F}}_{\text{F}}(\mathbf{p})$ yields the D-optimal input signal defined by

$$\omega_{\text{D}} = \begin{cases} \dfrac{1}{\sqrt{\tau^2 - 2a^2}} & \text{if } 2a^2 < \tau^2 \\ \infty & \text{otherwise.} \end{cases}$$

When a tends to 0, the noise becomes wide-band, and the D-optimal angular frequency is $\omega_{\text{D}} = 1/\tau$. This is intuitively appealing: for very low frequencies, the magnitude of the output does not depend on τ, whereas for very high frequencies the signal is buried in the noise. ◊

REMARKS 6.8

— A sequence of independent random variables is necessarily white noise. In the continuous-time case, such a signal should have infinite variance, which has no physical meaning. A reasonable assumption is that the continuous-time signal ε is white (has a flat power spectral density) over a frequency range *large enough* for the correlation between the $\varepsilon(t)$'s to have negligible effects on the measured outputs.

— Throughout this section, $u(t)$ has been assumed independent of $\varepsilon(t)$. This is no longer true if the input signal is obtained by feedback of the output. Another expression must then be calculated for the matrix $\mathbf{F}(\mathbf{p})$, which leads to the following results (Goodwin and Payne, 1977; Gustavsson, Ljung and Söderström, 1981; Gevers and Ljung, 1985, 1986): when F and G have no common parameters and the power constraint is on the input, the optimal input signal is obtained without feeding back the output; when the power constraint is on the output, feedback is generally useful (minimum-variance control).

— More recent results (Hjalmarsson, Gevers and De Bruyne, 1996) show that under rather general conditions, if the purpose of the identification is the design of a controller from the identified model, the experiment should preferably be performed in closed loop. ◊

6.3.3 Simultaneous choice of inputs and sampling times

When the input is parametric, the components of the vector Ξ that characterizes the experiment may consist of the n_t sampling times and the parameters defining the input signal, as in Section 6.3.2.1 (Example 6.7).

When the input signal is nonparametric, the construction of an optimal non-uniform sampling schedule (*i.e.* with $T_k = t_{k+1} - t_k$ not constant), although theoretically feasible (Goodwin and Payne, 1977), is rather difficult, especially off-line. An extension of the sequential approach described in Section 6.3.2.2 for the determination of the input $u(t_k)$ makes it possible to choose $u(t_k)$ and T_k simultaneously on-line (Goodwin, Zarrop and Payne, 1974; Goodwin and Payne, 1977).

Consider now the case of uniform sampling ($T_k = T$ constant) for a Box-Jenkins structure. The whole matrix $\mathbf{F}(\mathbf{p})$ is now influenced by the choice of the sampling period, even if F and G have no common parameters (compare with Section 6.3.2.2). We shall, however, assume that the spectral characteristics of the measurement noise are known. Consider then a system with the state-space representation

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t),$$
$$y(t) = \mathbf{c}^{\mathrm{T}}\mathbf{x}(t) + du(t) + \eta(t),$$

where $\mathbf{x}(t)$ is the state vector, $u(t)$ the (scalar) input, $y(t)$ the (scalar) measured output, and $\eta(t)$ a Gaussian stationary coloured noise, with known power spectral density $\psi(\omega)$. Note that $\eta(t)$ can be considered as the result of filtering a Gaussian white noise $\varepsilon(t)$ with unit spectral density by a filter with transfer function $\psi^{1/2}(j\omega)$. In the frequency domain, the transfer function of the deterministic part of the model is

$$F(j\omega, \mathbf{p}) = \mathbf{c}^{\mathrm{T}}(j\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d.$$

The transfer function $G(j\omega)$ associated with the random part does not depend on the parameters to be estimated. Let the number of measurements tend to infinity. The average Fisher information matrix per unit time is then

$$\overline{\mathbf{F}}(\mathbf{p}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{\partial F(j\omega, \mathbf{p})}{\partial \mathbf{p}} \psi^{-1}(\omega) \frac{\partial F(-j\omega, \mathbf{p})}{\partial \mathbf{p}^{\mathrm{T}}} p_{\mathrm{u}}(\omega)\mathrm{d}\omega,$$

where $p_{\mathrm{u}}(\omega)$ is the power spectral density of the input signal. Again, inputs with unit average power are considered:

$$P_{\mathrm{u}} = \frac{1}{2\pi} \int_{-\infty}^{\infty} p_{\mathrm{u}}(\omega)\mathrm{d}\omega = 1.$$

If the bandwidth of the input signal is restricted to ω_{u}, *i.e.* $\omega \in [-\omega_{\mathrm{u}}, \omega_{\mathrm{u}}]$, the matrix $\overline{\mathbf{F}}(\mathbf{p})$ is given by

$$\overline{\mathbf{F}}(\mathbf{p}) = \frac{1}{2\pi} \int_{-\omega_u}^{\omega_u} \frac{\partial F(j\omega, \mathbf{p})}{\partial \mathbf{p}} \psi^{-1}(\omega) \frac{\partial F(-j\omega, \mathbf{p})}{\partial \mathbf{p}^T} p_u(\omega) d\omega.$$

Assume that the output is sampled at a frequency $\omega_s/2\pi$ larger than Nyquist's lower bound, *i.e.* $\omega_s > 2\omega_u$. Then the deterministic part of the signal (due to u) is not altered, but aliasing occurs for the random part due to η. The power spectral density of the sampled noise is given by

$$\psi_s(\omega) = \psi(\omega) + \sum_{k=1}^{\infty} \psi(k\omega_s + \omega) + \psi(k\omega_s - \omega), \;\; \omega \in [-\frac{\omega_s}{2}, \frac{\omega_s}{2}],$$

so $\psi_s(\omega) \geq \psi(\omega)$. The Fisher information matrix $\overline{\mathbf{F}}_s(\mathbf{p})$ after sampling is obtained by substituting $\psi_s(\omega)$ for $\psi(\omega)$ in $\overline{\mathbf{F}}(\mathbf{p})$. The difference $\overline{\mathbf{F}}(\mathbf{p}) - \overline{\mathbf{F}}_s(\mathbf{p})$ is non-negative definite, so sampling can only reduce the precision of the estimates. However, this effect can be avoided by introducing a suitable *presampling filter*, with a transfer function $F_f(j\omega)$ such that $|F_f(j\omega)| = 0$ for $\omega \notin [-\omega_s/2, \omega_s/2]$ and $F_f(j\omega)$ is invertible elsewhere (Goodwin and Payne, 1977). Indeed, the power spectral density $\psi_{Ff}(\omega)$ of the filtered noise then satisfies

$$\psi_{Ff}(\omega) = |F_f(j\omega)|^2 \psi(\omega),$$

and sampling does not produce aliasing of this filtered noise. The Fisher information matrix *after filtering and sampling* is thus

$$\overline{\mathbf{F}}_f(\mathbf{p}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{\partial F(j\omega, \mathbf{p})}{\partial \mathbf{p}} F_f(j\omega) \, [\psi_{Ff}(\omega)]^{-1} \, F_f(-j\omega) \frac{\partial F(-j\omega, \mathbf{p})}{\partial \mathbf{p}^T} p_u(\omega) d\omega,$$

so $\overline{\mathbf{F}}_f(\mathbf{p}) = \overline{\mathbf{F}}(\mathbf{p})$. Sampling at the frequency $\omega_s/2\pi$ produces $\omega_s/2\pi$ samples per second, and the average Fisher information matrix per sample is

$$\mathbf{F}_{ps}(\mathbf{p}) = \frac{2\pi}{\omega_s} \overline{\mathbf{F}}(\mathbf{p}).$$

The optimal sampling frequency is thus the lowest satisfying Shannon's condition (*i.e.* the lowest frequency at which there is no aliasing of the input signal u), $\omega_s = 2\omega_u$. The matrix $\mathbf{F}_{ps}(\mathbf{p})$ is then

$$\mathbf{F}_{ps}(\mathbf{p}) = \frac{1}{\omega_u} \int_0^{\omega_u} \mathrm{Re}[\frac{\partial F(j\omega, \mathbf{p})}{\partial \mathbf{p}} \psi^{-1}(\omega) \frac{\partial F(-j\omega, \mathbf{p})}{\partial \mathbf{p}^T}] \, p_u(\omega) d\omega.$$

A method for determining the optimal input spectrum is presented in (Zarrop, 1979).

REMARK 6.9

Choosing the *lowest* feasible sampling frequency might seem surprising. Note, however, that the criterion relies on the average Fisher information matrix per sample. For a given number of samples, this policy thus yields the largest possible duration for the experiment. ◊

EXAMPLE 6.12

Consider again the system of Example 6.11, with transfer function

$$F(j\omega,\ \tau^*) = \frac{1}{1 + j\tau^*\omega},$$

and measurement noise $\eta(t)$ assumed to be wide-band, *i.e.* $\psi(\omega) = 1$. We want to determine the D-optimal angular frequency ω_D of a single sinusoid in order to estimate τ. The average Fisher information matrix per unit of time is

$$\overline{\mathbf{F}}(\mathbf{p}) = \frac{\omega^2}{(1 + \tau^2\omega^2)^2}.$$

When the highest frequency in the input signal is $\omega_u/(2\pi)$, sampling must be such that $\omega_s = 2\omega_u$ and the presampling filter must remove all components at frequencies above $\omega_u/(2\pi)$. For a sinusoid at frequency $\omega_u/(2\pi)$, the average Fisher information matrix per sample is

$$\mathbf{F}_{ps}(\mathbf{p}) = \frac{\pi\omega_u}{(1 + \tau^2\omega_u{}^2)^2}.$$

The angular frequency of the D-optimal input is then $\omega_D = 1/(\tau\sqrt{3})$, and the sampling period is $T_D = 2\pi/\omega_{sD} = \sqrt{3}\pi\tau$. Recall that when sampling was not considered, the angular frequency of the D-optimal input was $\omega_D = 1/\tau$. By reducing the frequency of the input signal, one can increase the sampling period and thus the duration of the experiment. ◊

REMARK 6.10

The experiment can also be decomposed into subexperiments, each employing a sinusoid with unit power and frequency $\omega_{u_i}/(2\pi)$, with sampling period $T_i = \pi/\omega_{u_i}$. The average Fisher information matrix per sample for the whole experiment is then

$$\mathbf{F}_{ps}(\mathbf{p}) = \sum_{i=1}^{n_\omega} \mu_i \frac{\pi}{\omega_{u_i}} \overline{\mathbf{F}}(\mathbf{p},\ \omega_{u_i}),$$

where $\overline{\mathbf{F}}(\mathbf{p},\ \omega_{u_i})$ is $\overline{\mathbf{F}}(\mathbf{p})$ calculated for a sinusoidal input with unit power and frequency $\omega_{u_i}/2\pi$. The scalar μ_i indicates the fraction of the total number of observations taken in the ith subexperiment (Goodwin and Payne, 1977). ◊

6.4 Robust design

6.4.1 Limitations of local design

A non-LP model structure raises two important problems for experiment design. The first is that the determinant of the Fisher information matrix may be a very approximate measure of the size of the uncertainty region. The second is that this matrix now depends on the value assumed for the parameters. To illustrate these difficulties by comparing the LP and non-LP cases, we shall consider least-squares estimation from data corrupted by additive i.i.d. $\mathcal{N}(0, \sigma^2)$ random variables, with σ^2 known.

Consider first an LP model structure. For given experimental conditions Ξ, the expectation surface

$$\mathbb{S}_{\text{exp}} = \{\mathbf{y}^{\text{m}}(\Xi, \mathbf{p}) = \mathbf{R}(\Xi)\mathbf{p} \mid \mathbf{p} \in \mathbb{R}^{n_{\text{p}}}\}$$

is an n_{p}-dimensional hyperplane. The least-squares estimate is obtained by projecting the vector of observations $\mathbf{y}^{\text{s}}$ onto $\mathbb{S}_{\text{exp}}$, which gives $\mathbf{y}^{\text{m}}(\Xi, \hat{\mathbf{p}}_{\text{ls}})$. A confidence region at level α for the parameters is then (Section 5.1.1.1)

$$\mathbb{R}_{\text{l}}^{\alpha} = \{\mathbf{p} \in \mathbb{R}^{n_{\text{p}}} \mid \|\mathbf{y}^{\text{m}}(\Xi, \mathbf{p}) - \mathbf{y}^{\text{s}}\|_2^2 \leq \sigma^2\chi_{\alpha}^2(n_{\text{t}})\},$$

where $\chi_{\alpha}^2(n_{\text{t}})$ has probability α of being smaller than a random variable with a χ^2 distribution with n_{t} degrees of freedom. The set

$$\mathbb{S}(\mathbf{y}^{\text{s}}) = \{\mathbf{y} \in \mathbb{R}^{n_{\text{t}}} \mid \|\mathbf{y} - \mathbf{y}^{\text{s}}\|_2^2 \leq \sigma^2\chi_{\alpha}^2(n_{\text{t}})\}$$

is a ball centred at $\mathbf{y}^{\text{s}}$. Its intersection with $\mathbb{S}_{\text{exp}}$ is an n_{p}-dimensional ball, centred at $\mathbf{y}^{\text{m}}(\Xi, \hat{\mathbf{p}}_{\text{ls}})$. Since the parametrization is linear, $\mathbb{R}_{\text{l}}^{\alpha}$ is an ellipsoid centred at $\hat{\mathbf{p}}_{\text{ls}}$, with volume proportional to $\det^{-1/2}[\mathbf{R}^{\text{T}}(\Xi)\mathbf{R}(\Xi)]$. Maximizing $\det[\mathbf{R}^{\text{T}}(\Xi)\mathbf{R}(\Xi)]$, *i.e.* designing a D-optimal experiment, thus minimizes the volume of confidence regions for $\mathbf{p}$ (at any level α).

Consider now a non-LP model structure. The region $\mathbb{R}_{\text{l}}^{\alpha}$ is no longer an ellipsoid and may have any shape (see Figure 5.3). There are two reasons for this (Bates and Watts, 1980). First, $\mathbb{S}_{\text{exp}}$ is generally a curved surface (intrinsic curvature), so its intersection with $\mathbb{S}(\mathbf{y}^{\text{s}})$ is no longer a ball. Second, the parametrization is nonlinear (parametric curvature). Moreover, the shape and volume of $\mathbb{R}_{\text{l}}^{\alpha}$ depend on the location of $\mathbf{y}^{\text{m}}(\Xi, \hat{\mathbf{p}}_{\text{ls}})$ on $\mathbb{S}_{\text{exp}}$, *i.e.* on the value of $\mathbf{y}^{\text{s}}$, and any experiment design based on the precision of the estimation will be *local*.

Consider first the issue of assessment of the size of the confidence regions in least-squares estimation. Approximating these confidence regions by ellipsoids amounts to linearizing the model (first-order expansion of $\mathbf{y}^{\text{m}}(\Xi, \mathbf{p})$), about some nominal value $\mathbf{p}^0$ of the parameters during the design phase or about $\hat{\mathbf{p}}_{\text{ls}}$ when estimation has taken place. This is equivalent to approximating the density of the least-squares estimator $\hat{\mathbf{p}}_{\text{ls}}(\mathbf{y}^{\text{s}})$ by $\mathcal{N}(\mathbf{p}^0, \mathbf{F}^{-1}(\mathbf{p}^0))$ or $\mathcal{N}(\hat{\mathbf{p}}_{\text{ls}}, \mathbf{F}^{-1}(\hat{\mathbf{p}}_{\text{ls}}))$.

Hamilton and Watts (1985) consider second-order approximation of the volume of (non-ellipsoidal) confidence regions for $\mathbf{p}$. However, the accuracy of the resulting approximation seems hard to evaluate. Vila (1986, 1990) uses a series expansion of a non-centred Fisher-Snedecor distribution and numerical integration to design experiments that minimize the volume of exact confidence regions.

We have seen in Section 5.3.3 how the (approximate) density $q_{\Xi}(\hat{\mathbf{p}}_{\text{ls}}|\mathbf{p}^*)$ of the least-squares estimator, or its marginal densities, could be used to assess the precision of the estimation. This density may differ greatly from the asymptotic normal density; see Examples 5.2 and 5.3. In such a case, the D-optimality cost function may be a very crude measure of the precision of the estimation. The mean-square error

$$j(\Xi) = \int \|\hat{\mathbf{p}}_{\text{ls}} - \mathbf{p}^*\|_2^2 q_{\Xi}(\hat{\mathbf{p}}_{\text{ls}}|\mathbf{p}^*)\mathrm{d}\hat{\mathbf{p}}_{\text{ls}}$$

is used in (Pázman and Pronzato, 1992a, 1992b) as a generalization of the A-optimal design cost function. However, when the integral is evaluated over a bounded domain $\mathbb{P}$ for $\hat{\mathbf{p}}_{\text{ls}}$, the optimal experiment is degenerate, *i.e.* has

$$\int_{\mathbb{P}} q_{\Xi}(\hat{\mathbf{p}}_{\text{ls}}|\mathbf{p}^*)\mathrm{d}\hat{\mathbf{p}}_{\text{ls}} = 0.$$

For that reason, a constrained estimator is used, given by

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \|\mathbf{y}^{\text{m}}(\Xi, \mathbf{p}) - \mathbf{y}^{\text{s}}\|_2^2 + 2\sigma^2 w(\mathbf{p}),$$

with $w(\mathbf{p})$ a suitable penalty function constraining $\hat{\mathbf{p}}$ to lie in a compact set $\mathbb{P}$. A possible choice when $\mathbb{P}$ is the orthotope

$$\mathbb{P} = [p_1^{\text{m}}, p_1^{\text{M}}] \times \ldots \times [p_{n_\text{p}}^{\text{m}}, p_{n_\text{p}}^{\text{M}}],$$

is for instance

$$w(\mathbf{p}) = \sum_{i=1}^{n_\text{p}} w_i(p_i)$$

where:

$$\begin{cases} w_i(p_i) = 0 & \text{if } p_i \in [p_i^{\text{m}} + \Delta_i, p_i^{\text{M}} - \Delta_i], \\ \partial^2 w_i(p_i)/\partial p_i{}^2 > 0 & \text{if } p_i \notin [p_i^{\text{m}} + \Delta_i, p_i^{\text{M}} - \Delta_i], \\ \lim_{p_i \to p_i^{\text{m}}} w_i(p_i) = \lim_{p_i \to p_i^{\text{M}}} w_i(p_i) = \infty, & \end{cases}$$

and Δ_i is small compared to $p_i^{\text{M}} - p_i^{\text{m}}$. The density $q_{\Xi}(\hat{\mathbf{p}}|\mathbf{p}^*)$ of the estimator $\hat{\mathbf{p}}$ is then given by (Pázman and Pronzato, 1992a, 1992b)

$$q_{\Xi}(\hat{\mathbf{p}}|\mathbf{p}^*) = \frac{\det\left[\mathbf{Q}_{\Xi}(\hat{\mathbf{p}},\ \mathbf{p}^*) - \dfrac{1}{\sigma^2}\displaystyle\sum_{i=1}^{n_t} u_{\Xi_i}(\hat{\mathbf{p}})\ \dfrac{\partial^2 y_{\mathrm{m}}(\boldsymbol{\xi}^i,\ \mathbf{p})}{\partial\mathbf{p}\partial\mathbf{p}^{\mathrm{T}}}\Big|_{\hat{\mathbf{p}}} + \dfrac{\partial^2 w(\mathbf{p})}{\partial\mathbf{p}\partial\mathbf{p}^{\mathrm{T}}}\Big|_{\hat{\mathbf{p}}}\right]}{(2\pi)^{n_{\mathrm{p}}/2}\det^{1/2}\mathbf{F}(\hat{\mathbf{p}},\ \Xi)}$$

$$\times \exp\{-\frac{1}{2\sigma^2}\|\boldsymbol{\Pi}_{\Xi}(\hat{\mathbf{p}})[\mathbf{y}^{\mathrm{m}}(\Xi,\ \hat{\mathbf{p}}) - \mathbf{y}^{\mathrm{m}}(\Xi,\ \mathbf{p}^*)] + \mathbf{u}_{\Xi}(\hat{\mathbf{p}})\|_2^2\},$$

with

$$\mathbf{u}_{\Xi}(\mathbf{p}) = \frac{\partial\mathbf{y}^{\mathrm{m}}(\Xi,\ \mathbf{p})}{\partial\mathbf{p}^{\mathrm{T}}}\mathbf{F}^{-1}(\mathbf{p},\ \Xi)\frac{\partial w(\mathbf{p})}{\partial\mathbf{p}},$$

and $\mathbf{Q}_{\Xi}(\hat{\mathbf{p}},\ \mathbf{p}^*)$ and $\boldsymbol{\Pi}_{\Xi}(\hat{\mathbf{p}})$ as in Section 5.3.3. Note that the density of the maximum *a posteriori* estimator (Section 3.5.1) for measurement noise distributed $\mathcal{N}(0,\ \sigma^2)$ is obtained by taking $w(\mathbf{p}) = -\ln\ \pi_{\mathrm{p}}(\mathbf{p})$. Evaluation of the mean-square error $j(\Xi)$ corresponds to integration with respect to $\hat{\mathbf{p}}$. However, a stochastic-approximation algorithm permits optimization of $j(\Xi)$ without evaluating any integral.

Another design criterion based on the density $q_{\Xi}(\hat{\mathbf{p}}_{\mathrm{ls}}|\mathbf{p}^*)$ has been suggested by the following arguments. Designing a D-optimal experiment for an LP model structure defined by $\mathbf{y}^{\mathrm{m}}(\Xi,\ \mathbf{p}) = \mathbf{R}(\Xi)\mathbf{p}$ corresponds to minimizing the (Shannon) entropy $\hbar$ of the density of the estimator, given by

$$\hbar = -\frac{1}{2}\ln\det\left[\frac{1}{\sigma^2}\mathbf{R}^{\mathrm{T}}(\Xi)\mathbf{R}(\Xi)\right] + \frac{n_{\mathrm{p}}}{2}\ln 2\pi + \frac{n_{\mathrm{p}}}{2}.$$

D-optimal design for a non-LP model structure can thus be interpreted as minimization of a first-order approximation of the entropy of the density of the estimator. (The true density is approximated by the asymptotic normal density.) A second-order approximation based on $q_{\Xi}(\hat{\mathbf{p}}_{\mathrm{ls}}|\mathbf{p}^*)$ is suggested in (Pronzato and Pázman, 1994b).

Although less approximate than those based on the Fisher information matrix, these approaches remain local. The optimal experiment still depends on the value of the parameters to be estimated, which is unknown before the experiment. A first way of facing this problem is to design the experiments sequentially.

6.4.2 Sequential design

Experimentation and estimation steps are alternated (Chernoff, 1975), as indicated in Figure 6.6. Such an approach is natural when experimentation is thought of as on-line control of the system, with real-time estimation of the parameters and characterization of the precision achieved, as in Section 6.3.2.2. Even when estimation is off-line, and provided the observations from the experiment described by Ξ^i make $\mathbf{p}$ identifiable, one may wish to estimate $\mathbf{p}$ in the following way:

— estimate $\hat{\mathbf{p}}(i)$ from observations $\mathbf{y}(\Xi^i)$, for $i = 1,\ \dots,\ k$,
— estimate $\hat{\mathbf{p}}^k$ as the average of the $\hat{\mathbf{p}}(i)$'s:

$$\hat{\mathbf{p}}^k = \bar{\mathbf{p}}^k = \frac{1}{k}\sum_{i=1}^{k} \hat{\mathbf{p}}(i).$$

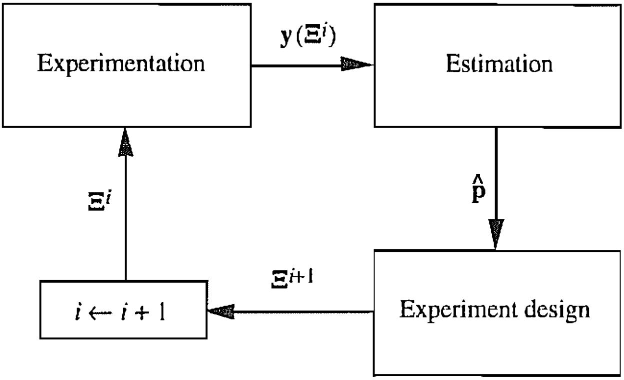


Figure 6.6. Sequential design

For non-LP model structures, each estimate $\hat{\mathbf{p}}(i)$ is generally biased; see (Box, 1971) for an approximation of the bias of the least-squares estimator. Moreover, $\bar{\mathbf{p}}^k$ usually does not converge to $\mathbf{p}^*$ as k tends to infinity. This approach should therefore *not* be used. To guarantee convergence of $\hat{\mathbf{p}}^k$ to $\mathbf{p}^*$, the estimation of $\hat{\mathbf{p}}^k$ should make use of all previous observations (*i.e.* $\mathbf{y}(\Xi^1), \ldots, \mathbf{y}(\Xi^k)$). The designed experiment will then tend to the optimal experiment for $\mathbf{p}^*$.

Often, however, the repetition of experiments on a single process is impossible (biology, destructive experiments...), but it is possible to experiment on a population of processes (or individuals), each experiment being performed on a new individual; see, *e.g.*, (D'Argenio, 1981). No single true value $\mathbf{p}^*$ then exists for $\mathbf{p}$, and each individual may be considered as having its own true parameter value $\mathbf{p}^{i*}$. One may then wish to design experiments converging to the optimal experiment for the average value of the $\mathbf{p}^{i*}$'s in the population (or, see Section 6.4.3, to the average optimal experiment for the population). A characteristic of the population (the mean of the $\mathbf{p}^{i*}$'s or their distribution) should then be estimated (possibly on-line) from the observations. Methods suggested in Section 3.3.4 may be used for that purpose.

REMARKS 6.11

— The study of the convergence properties of sequential design policies is often very complicated, and far beyond the scope of this book. One may refer for instance to (Ford and Silvey, 1980; Ford, Titterington and Wu, 1985; Wu, 1985; Müller and Pötscher, 1992) to get an idea of the difficulties.

— One may wish to determine the optimal sequential design procedure for a given design criterion and with the number of experiment steps fixed *a priori*. This is a stochastic dynamic programming problem, very difficult to solve. Examples with two experiment steps are presented in (Zacks, 1977; Pronzato, Walter and Kulcsâr, 1993; Kulcsâr, Pronzato and Walter, 1994). A classification of sequential design policies, considered as control policies for dynamic systems, is given in

(Bayard and Schumitzky, 1990; Pronzato, Walter and Kulcsár, 1993). A suboptimal closed-loop approach (dual control), based on approximation of the posterior density of the parameters, is suggested in (Kulcsár, Pronzato and Walter, 1995, 1996). ◊

In many situations, repetition of experiments is impossible, even on several individuals or processes, and a single (one-shot) experiment must be designed. Moreover, even if design is sequential, each design step should make use of all information available. Non-sequential design approaches which determine a single experiment taking all the prior uncertainty in the parameters to be estimated into account are thus of special importance. Two types of robust design procedure will be considered. They differ in how the prior information is characterized, and by the importance attached to the risk of designing an experiment badly suited to some rare parameter values. Only robustness with respect to the parameters **p** of the deterministic model will be considered. Robustness with respect to nuisance parameters, *e.g.*, those present in the distribution of the measurement noise, could be handled in the same way (Schulz and Endrenyi, 1983). In that case, a robust extension of D_s-optimality could also be used.

6.4.3 Average optimality

This approach relies on a probabilistic description of the prior uncertainty in **p**, characterized by a prior distribution $\pi_{\mathrm{p}}(\mathbf{p})$. Note that if this distribution is reliable prior information, Bayesian estimators (Section 3.5) are more appropriate than the maximum-likelihood estimator. However, here we consider distributions which carry little information, and we are mainly concerned with quantifying the lack of reliability of the prior nominal value for **p**. In this context, it is natural to let the observations speak by themselves and use the maximum-likelihood estimator. The distribution $\pi_{\mathrm{p}}(\mathbf{p})$ may have been inferred from previous observations collected on similar processes or individuals in a population (Section 3.3.4).

6.4.3.1 Criteria

Classical criteria lead to optimization of a scalar function of the Fisher information matrix. We shall only consider cost functions related to D-optimality, but other cost functions could be treated similarly. Using the prior distribution $\pi_{\mathrm{p}}(\mathbf{p})$ makes it possible to remove the dependence on **p** by considering the expectation of the original cost function. Note that whereas cost functions $-\det \mathbf{F}(\mathbf{p}, \Xi)$, $-\ln \det \mathbf{F}(\mathbf{p}, \Xi)$ and $1/\det \mathbf{F}(\mathbf{p}, \Xi)$ lead to identical designs, the introduction of expectations makes these approaches different (Fedorov, 1980; Fedorov and Atkinson, 1988; Atkinson, 1992).

— *ED-optimal design* (Pronzato and Walter, 1985) maximizes

$$j_{\mathrm{ED}}(\Xi) = \underset{\mathbf{p}}{\mathrm{E}} \{\det \mathbf{F}(\mathbf{p}, \Xi)\}.$$

— *EID-optimal design* (Walter and Pronzato, 1987) minimizes

$$j_{\mathrm{EID}}(\Xi) = \mathop{\mathrm{E}}_{\mathbf{p}} \{1/\det \mathbf{F}(\mathbf{p}, \Xi)\}.$$

— *ELD-optimal design* (D'Argenio and Van Guilder, 1988; D'Argenio, 1990) maximizes

$$j_{\mathrm{ELD}}(\Xi) = \mathop{\mathrm{E}}_{\mathbf{p}} \{\ln \det \mathbf{F}(\mathbf{p}, \Xi)\}.$$

Averaging D-efficiency cost functions also allows new cost functions to be defined:

— for *ED-efficiency* (or *EDE-optimality*), maximize

$$j_{\mathrm{EDE}}(\Xi) = \mathop{\mathrm{E}}_{\mathbf{p}} [j_{\mathrm{DE}}(\mathbf{p}, \Xi)],$$

where j_{DE} is defined in Section 6.1,

— for *EID-efficiency* (or *EIDE-optimality*), minimize

$$j_{\mathrm{EIDE}}(\Xi) = \mathop{\mathrm{E}}_{\mathbf{p}} \{1/j_{\mathrm{DE}}(\mathbf{p}, \Xi)\}.$$

The choice of a cost function may take account of the following facts.

— D-efficiency is relative: for each value of $\mathbf{p}$, the performance of the experiment Ξ is scaled by the performance achievable if $\mathbf{p}$ were known. Approaches relying on j_{EDE} and j_{EIDE} thus favour those parameters whose estimation is difficult. Depending on the circumstances, this may be an advantage or a drawback.

— When optimizing j_{EDE} or j_{EIDE}, each evaluation of the D-efficiency cost function for a given value $\mathbf{p}$ of the parameters requires the determination of a D-optimal experiment for $\mathbf{p}$. This optimization thus requires much more computation than that of j_{ED}, j_{EID} or j_{ELD}.

— ED-optimality does not seem a suitable measure to characterize the average uncertainty in the parameters (Walter and Pronzato, 1987), so EID-optimality should be preferred.

— ELD-optimality can be justified by information-theoretic arguments: an ELD-optimal experiment maximizes the prior expectation of the information provided by the experiment, under the assumption that the prior information is negligible compared with that yielded by the experiment (D'Argenio, 1990).

— Finally, an EID-optimal experiment depends on the parametrization of the model structure, whereas an ELD-optimal experiment does not. Indeed, consider a non-singular reparametrization $\tilde{\mathbf{p}}(\mathbf{p})$ (*i.e.* with $\det \partial\tilde{\mathbf{p}}/\partial\mathbf{p}^{\mathrm{T}} \neq 0$), independent of Ξ. We have

$$\det \mathbf{F}(\tilde{\mathbf{p}}, \Xi) = \det \mathbf{F}(\mathbf{p}, \Xi) \left(\det \frac{\partial\tilde{\mathbf{p}}}{\partial\mathbf{p}^{\mathrm{T}}}\right)^{-2}$$

and $\mathrm{E}_{\mathbf{p}}\{1/\det \mathbf{F}(\mathbf{p}, \Xi)\}$ generally differs from $\mathrm{E}_{\mathbf{p}}\{1/\det \mathbf{F}(\tilde{\mathbf{p}}, \Xi)\}$. However,

$$\underset{\mathbf{p}}{\mathrm{E}}\{\ln\det\mathbf{F}(\tilde{\mathbf{p}},\Xi)\} = \underset{\mathbf{p}}{\mathrm{E}}\{\ln\det\mathbf{F}(\mathbf{p},\Xi)\} - 2\,\underset{\mathbf{p}}{\mathrm{E}}\{\ln\det\frac{\partial\tilde{\mathbf{p}}}{\partial\mathbf{p}^{\mathrm{T}}}\}$$

and the last term is independent of Ξ.

Pronzato *et al.* (1989) and Atkinson *et al.* (1993) compare these approaches on pharmacokinetic examples. A reasonable policy might be to determine the optimal experiments for various criteria, then choose an experiment which is optimal for one particular criterion and only slightly suboptimal for the others.

EXAMPLE 6.2 (continued)

Consider again the system:

$$y(t) = p_1^* \exp(-p_2^* t) + \varepsilon(t),$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, \sigma^2)$. We are looking for two optimal sampling times, to estimate p_1^* and p_2^*. The model structure is nonlinear in p_2, and the D-optimal experiment for a nominal value p_2^0 is

$$\Xi_{\mathrm{D}} = (0,\ 1/p_2^0)^{\mathrm{T}}.$$

Consider now a prior density $\pi_{\mathrm{p}}(p_2)$ to characterize the uncertainty in p_2^0. The EID-optimality cost function is

$$j_{\mathrm{EID}}(\Xi) = \frac{\sigma^4}{p_1^2(t_1 - t_2)^2} \int_{\mathbb{P}} \exp[2p_2(t_1 + t_2)]\ \pi_{\mathrm{p}}(p_2)\mathrm{d}p_2.$$

When $\pi_{\mathrm{p}}(p_2)$ is uniform over [1, 10], the EID-optimal experiment is

$$\Xi_{\mathrm{EID}} = (0,\ \ 0.139)^{\mathrm{T}}.$$

When $\pi_{\mathrm{p}}(p_2)$ is normal $\mathcal{N}(5.5,\ 1.5^2)$, the EID-optimal experiment is

$$\Xi_{\mathrm{EID}} = (0,\ \ 0.161)^{\mathrm{T}}.$$

The ELD-optimality cost function is

$$j_{\mathrm{ELD}}(\Xi) = -2\,\underset{p_2}{\mathrm{E}}\{p_2\}\ (t_1 + t_2)\ \ln\frac{p_1^2(t_2 - t_1)^2}{\sigma^4}$$

and, in this particular case, the ELD-optimal experiment coincides with the D-optimal experiment calculated for a nominal value equal to the prior mean $\mathrm{E}_{p_2}\{p_2\}$. For the two densities above, its numerical value is

$$\Xi_{\mathrm{ELD}} = (0,\ \ 0.182)^{\mathrm{T}}. \qquad \Diamond$$

In the example above, the optimal experiment is independent of the value (or density) of parameter p_1. This result can be generalized, as indicated by the following property (Pronzato and Walter, 1985). If the model structure is such that

$$y_m(t, \mathbf{p}) = \mathbf{r}^T(t, \mathbf{p}^{nl})\mathbf{p}^l, \text{ with } \mathbf{p} = \begin{bmatrix} \mathbf{p}^l \\ \mathbf{p}^{nl} \end{bmatrix},$$

where $r_i(t, \mathbf{p}^{nl})$ depends only on the ith component of $\mathbf{p}^{nl}$, if the measurement noise corresponds to an i.i.d. sequence whose distribution is independent of $\mathbf{p}$, and if the linear parameters $\mathbf{p}^l$ are distributed independently of the nonlinear parameters $\mathbf{p}^{nl}$, then the optimal experiment for the criteria above ($j_{ED}, j_{EID}, j_{ELD}, j_{EDE}$ and j_{EIDE}) is independent of the distribution of the parameters $\mathbf{p}^l$. For instance, all entries of $\mathbf{p}^l$ can be set equal to one.

Except in some very simple situations, an optimal experiment cannot be found analytically, and numerical procedures are required.

6.4.3.2 Algorithms

Exact design. A first approach uses one of the local design methods presented in Section 6.2.1, with some general-purpose nonlinear-programming algorithm or a specific algorithm for exact design. At each iteration, an expected value of a local cost function has to be evaluated. Such an approach can be employed when the prior distribution for $\mathbf{p}$ is discrete (D'Argenio and Van Guilder, 1988; Pronzato *et al.*, 1989), but is very slow when the prior distribution is a density. Stochastic approximation then allows a cost function like

$$j_E(\Xi) = \underset{\mathbf{p}}{\mathrm{E}} \{j(\mathbf{p}, \Xi)\}$$

to be optimized without having to evaluate expectations (*i.e.* integrals); see Section 4.3.8. The simplest version is the stochastic gradient algorithm, which, for minimization, is

$$\Xi^{k+1} = \Xi^k - \lambda_k \frac{\partial j(\mathbf{p}^k, \Xi)}{\partial \Xi}\bigg|_{\Xi = \Xi^k}.$$

It is a gradient algorithm (Section 4.3.3.1) for the minimization of $j(\mathbf{p}, \Xi)$, modified in that, at each iteration k, a value $\mathbf{p}^k$ is randomly generated according to $\pi_p(\mathbf{p})$. The sequence of scalar steps λ_k must satisfy

$$\lambda_k \geq 0, \quad \sum_{k=0}^{\infty} \lambda_k = \infty, \quad \sum_{k=0}^{\infty} \lambda_k^2 < \infty,$$

and the most popular choice is the harmonic sequence

$$\lambda_k = \frac{\alpha}{k+1}, \quad \alpha > 0.$$

Convergence is accelerated if λ_k is reduced only when the angle between two successive gradients is larger than $\pi/2$ (Saridis, 1974). The speed of convergence is very sensitive to the choice of the scalar α in the sequence λ_k. Componentwise normalization of the gradient allows easier choice of a suitable value for α. The algorithm thus modified becomes

$$\Xi^{k+1} = \Xi^k - \lambda_k \, \boldsymbol{\Lambda}_k \frac{\partial j(\mathbf{p}^k, \Xi)}{\partial \Xi} |_{\Xi = \Xi^k},$$

where $\boldsymbol{\Lambda}_k$ is a diagonal matrix, the ith diagonal entry of which is

$$\Lambda_{k_{ii}} = \frac{\Xi_{i_{\max}} - \Xi_{i_{\min}}}{\left[\frac{1}{k} \sum_{n=1}^{k} \left(\frac{\partial j(\mathbf{p}^n, \Xi)}{\partial \Xi_i} |_{\Xi = \Xi^n}\right)^2\right]^{1/2}},$$

where $\Xi_{i_{\max}}$ and $\Xi_{i_{\min}}$ are upper and lower bounds on the possible values of Ξ_i. If $\lambda_k = \alpha/(k+1)$, this implies

$$\Xi_i^1 - \Xi_i^0 = \pm\, \alpha(\Xi_{i_{\max}} - \Xi_{i_{\min}}).$$

The scalar α is then the relative length of the first step. A typical choice is $\alpha = 0.1$. Examples of application are presented in (Pronzato and Walter, 1985; Walter and Pronzato, 1987). Note that convergence to a global optimum is not guaranteed.

Approximate design. The equivalence theorem of Section 6.2.2.3 can be extended to average-optimal design; see (Atkinson, 1992) for theorems applying to various optimality criteria. For ELD-optimality, for instance, the following theorem is available.

EQUIVALENCE THEOREM

The following properties are equivalent:

— the design measure m_{ELD} is ELD-optimal,
— $\max_{\xi \in \xi} \mathrm{E}_{\mathbf{p}}\{d(\xi, m_{\mathrm{ELD}})\} = n_{\mathrm{p}}$,
— m_{ELD} minimizes $\max_{\xi \in \xi} \mathrm{E}_{\mathbf{p}}\{d(\xi, m)\}$, with

$$d(\xi, m) = \frac{1}{w(\xi)} \mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi, \mathbf{p}) \mathbf{F}_{\mathrm{m}}^{-1}(\mathbf{p}, m) \mathbf{s}_{\mathrm{y}}(\xi, \mathbf{p}). \qquad \Diamond$$

An algorithm similar to that of Fedorov and Wynn (Section 6.2.2.4) can thus be used, $d(\xi, m)$ simply being replaced by its expected value $\mathrm{E}_{\mathbf{p}}\{d(\xi, m)\}$ (Chaloner and Larntz, 1986, 1988). Global convergence to an ELD-optimal design measure is guaranteed, but, here again, calculations will be extremely heavy if the prior distribution for $\mathbf{p}$ is not discrete.

REMARK 6.12

Average optimality can be used in sequential design too. Each experimentation step then makes density $\pi_p(\mathbf{p})$ in the next step more accurate. Two cases must be distinguished, depending on whether the observations are performed on a single process or on different processes (or individuals) drawn from a population; see, *e.g.*, (Pronzato, Walter and Kulcsár, 1993). In the latter case, maximum-likelihood estimation of $\pi_p(\mathbf{p})$ can be performed with the methods of Section 3.3.4 (using all data obtained so far). The recursive approach presented in (Mentré, 1984; Mentré, Mallet and Steimer, 1988) is then particularly attractive. Note that optimal design for the estimation of $\pi_p(\mathbf{p})$ has received little attention in spite of its importance (Mallet, 1983; Mallet and Mentré, 1988; Mentré *et al.*, 1995). ◊

6.4.4 Minimax optimality

Sometimes, the best experiment in the worst circumstances should be preferred to the best one on average. This depends on the importance attached to some (unlikely) parameter values whose estimation with an average-optimal experiment might be very inaccurate. Minimax optimal design requires the definition of a set $\mathbb{P}$ of prior admissible values for $\mathbf{p}$.

6.4.4.1 Criteria

We shall only consider criteria based on D-optimality (Section 6.1). *MMD-optimal design* (Pronzato and Walter, 1988) maximizes

$$j_{\mathrm{MMD}}(\Xi) = \min_{\mathbf{p}\in\mathbb{P}} \det \mathbf{F}(\mathbf{p}, \Xi).$$

MMDE-optimal design (Landaw, 1984; D'Argenio and Van Guilder, 1988) maximizes

$$j_{\mathrm{MMDE}}(\Xi) = \min_{\mathbf{p}\in\mathbb{P}} j_{\mathrm{DE}}(\mathbf{p}, \Xi),$$

where $j_{\mathrm{DE}}(\mathbf{p}, \Xi)$ is the D-efficiency cost function. An MMDE-optimal experiment does not depend on any regular reparametrization of the model structure (independent of Ξ), whereas an MMD-optimal experiment generally does. On the other hand, optimization of j_{MMDE} requires the determination of a D-optimal experiment associated with $\mathbf{p}$ for each evaluation of $j_{\mathrm{DE}}(\mathbf{p}, \Xi)$. This optimization will thus require more computation than that of j_{MMD}. Normalization by $\det \mathbf{F}(\mathbf{p}, \Xi_{\mathrm{D}})$ in the expression of D-efficiency favours those parameters whose estimation is difficult. This may prove to be an advantage or a drawback (Pronzato and Walter, 1988).

EXAMPLE 6.2 (continued)

Consider again the system:

$$y(t) = p_1^* \exp(-p_2^* t) + \varepsilon(t),$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, \sigma^2)$. We search for two sampling times to estimate the parameters p_1^* and p_2^*, when the prior admissible values for p_2 belong to

$$\mathbb{P}_2 = [p_{2\min}, p_{2\max}].$$

The MMD-optimal experiment then coincides with the D-optimal one for $p_2 = p_{2\max}$:

$$\Xi_{\text{MMD}} = \left(0, \ \frac{1}{p_{2\max}}\right)^{\text{T}}.$$

The MMDE-optimal experiment is

$$\Xi_{\text{MMDE}} = \left(0, \ \frac{\ln p_{2\max} - \ln p_{2\min}}{p_{2\max} - p_{2\min}}\right)^{\text{T}}.$$ ◊

EXAMPLE 6.11 (continued)

Consider again the transfer function

$$F(j\omega, \tau^*) = \frac{1}{1 + j\tau^*\omega},$$

and assume the measurement noise is wide-band. We wish to determine the input to be applied to the system in order to estimate τ as accurately as possible. A sinusoidal input, with angular frequency $\omega_{\text{D}} = 1/\tau^*$ which strongly depends on τ^*, is then D-optimal. If the admissible values for τ are in the interval $\mathbb{P} = [\tau_{\min}, \tau_{\max}]$, the MMD-optimality cost function for a single sinusoid is

$$j_{\text{MMD}}(\omega) = \min_{\tau\in\mathbb{P}} \frac{\omega^2}{(1 + \tau^2\omega^2)^2},$$

and the MMD-optimal angular frequency is $1/\tau_{\max}$, which coincides with the D-optimal angular frequency for $\tau = \tau_{\max}$. The MMDE-optimality cost function is

$$j_{\text{MMDE}}(\omega) = \min_{\tau\in\mathbb{P}} \frac{4\tau^2\omega^2}{(1 + \tau^2\omega^2)^2},$$

and the MMDE-optimal angular frequency is $(\tau_{\min}\tau_{\max})^{-1/2}$, *i.e.* the geometric mean of the D-optimal angular frequencies for $\tau = \tau_{\min}$ and $\tau = \tau_{\max}$. ◊

As for average optimal design, if the model structure is such that

$$y_{\text{m}}(t, \mathbf{p}) = \mathbf{r}^{\text{T}}(t, \mathbf{p}^{\text{nl}})\, \mathbf{p}^{\text{l}},$$

where $r_i(t, \mathbf{p}^{\text{nl}})$ depends only on the ith component of $\mathbf{p}^{\text{nl}}$, and if the measurement noise corresponds to an i.i.d. sequence with distribution independent of $\mathbf{p}$, then the MMD- and MMDE-optimal experiments are independent of the value of the linear parameters $\mathbf{p}^{\mathrm{l}}$, provided the admissible domain for $\mathbf{p}^{\mathrm{nl}}$ is independent of $\mathbf{p}^{\mathrm{l}}$ (Pronzato and Walter, 1988). In particular, one can then set all components of $\mathbf{p}^{\mathrm{l}}$ to one.

In some situations, the minimax optimal experiment coincides with a D-optimal experiment for a particular value of the model parameters, which can be determined *a priori*. This greatly facilitates computation of the minimax optimal experiment. This is the case for structures defined by sums of exponentials when MMD-optimal design is used. Such structures play an important role, since they include the impulse responses of many systems. Consider a model structure defined by the response

$$y_{\mathrm{m}}(\xi^i, \mathbf{p}) = \sum_{n=1}^{n_{\mathrm{p}}/2} p_n^{\mathrm{l}} \exp(-p_n^{\mathrm{nl}} \xi^i)$$

and assume that the observations are corrupted by additive noise which corresponds to an i.i.d. sequence distributed independently of $\mathbf{p}$. The scalar ξ^i characterizes the experimental conditions for the ith observation; it might, for instance, be the ith sampling time. Assume that the prior admissible set for the vector $\mathbf{p}^{\mathrm{nl}}$ of nonlinear parameters is

$$\mathbb{P}^{\mathrm{nl}} = \{\mathbf{p}^{\mathrm{nl}} \in \mathbb{R}^{n_{\mathrm{p}}/2} \mid p_1^{\mathrm{nl}} \leq p_{\max}^{\mathrm{nl}}, p_i^{\mathrm{nl}} - p_{i+1}^{\mathrm{nl}} \geq \delta_i, i = 1, \dots, (n_{\mathrm{p}}/2) - 1\},$$

where $p_{\max}^{\mathrm{nl}}$ and the δ_i's are known. Then, from the property above, the MMD-optimal experiment is independent of the value of $\mathbf{p}^{\mathrm{l}}$, and, from (Melas, 1978), it coincides with the D-optimal experiment calculated for

$$\mathbf{p}^{\mathrm{nl}} = [p_{\max}^{\mathrm{nl}}, p_{\max}^{\mathrm{nl}} - \delta_1, p_{\max}^{\mathrm{nl}} - (\delta_1 + \delta_2), \dots, p_{\max}^{\mathrm{nl}} - \sum_{i=1}^{(n_{\mathrm{p}}/2)-1} \delta_i]^{\mathrm{T}}.$$

When the minimax-optimal experiment cannot be found analytically and does not coincide with a particular D-optimal experiment, specific optimization algorithms must be used.

6.4.4.2 Algorithms

Exact design: relaxation algorithm. When the prior admissible set $\mathbb{P}$ for $\mathbf{p}$ is finite, the minimization with respect to $\mathbf{p}$ in j_{MMD} and j_{MMDE} may in principle be carried out by exhaustive search over all values of $\mathbf{p}$ in $\mathbb{P}$. However, this is possible only if the number of elements in $\mathbb{P}$ is small enough.

Consider the general case, finding Ξ_{MM} (from some feasible set $\mathbb{E}$) to maximize

$$j_{\mathrm{MM}}(\Xi) = \min_{\mathbf{p} \in \mathbb{P}} j(\mathbf{p}, \Xi),$$

with $\mathbb{P}$ a given compact set. The most straightforward approach uses brute force and maximizes j_{MM} by a general-purpose nonlinear-programming algorithm, each evaluation of $j_{\mathrm{MM}}(\Xi)$ being the result of a minimization with respect to $\mathbf{p}$ using a second nonlinear programming algorithm. Obviously, this may require a huge amount of computation. Shimizu and Aiyoshi (1980) use relaxation to transform the problem into one where $\mathbb{P}$ is finite. First, the problem is redefined as finding Ξ_{MM} in $\mathbb{\Xi}$ that maximizes the scalar α under the constraints

$$j(\mathbf{p}, \Xi) \geq \alpha, \ \forall \ \mathbf{p} \in \mathbb{P}.$$

This optimization problem has an infinite number of constraints. The relaxation procedure amounts to introducing a finite number of them, one by one.

Step 1: Choose an initial value $\mathbf{p}^1$ in $\mathbb{P}$, and define a first set of representative values $\mathcal{R}^1 = \{\mathbf{p}^1\}$. Set $k = 1$.

Step 2: Find

$$\Xi^k = \arg \max_{\Xi \in \mathbb{\Xi}} \ \min_{\mathbf{p} \in \mathcal{R}^k} \ j(\mathbf{p}, \Xi).$$

Step 3: Find

$$\mathbf{p}^{k+1} = \arg \min_{\mathbf{p} \in \mathbb{P}} j(\mathbf{p}, \Xi^k).$$

Step 4: If $j(\mathbf{p}^{k+1}, \Xi^k) \geq \min_{\mathbf{p} \in \mathcal{R}^k} j(\mathbf{p}, \Xi^k) - \delta$, where δ is some positive tolerance, accept Ξ^k as an approximate solution of the problem. Else include $\mathbf{p}^{k+1}$ in $\mathcal{R}^k$, increment k by one and go to Step 2.

This algorithm stops after a finite number of steps, provided the following conditions are satisfied:

C1: $j(\Xi, \mathbf{p})$ is continuous with respect to $\mathbf{p}$ and continuously differentiable with respect to Ξ.

C2: $\mathbb{\Xi}$ is compact and such that $\mathbb{\Xi} = \{\Xi \mid c_i(\Xi) \leq 0, \ i = 1, \ldots, r\}$, with constraints c_i continuously differentiable with respect to Ξ.

C3: $\mathbb{P}$ is compact (and not empty).

These conditions are generally fulfilled for optimal-design problems. Note that when the algorithm is stopped before the stopping condition of Step 4 is satisfied, an approximate solution is obtained, satisfying a similar stopping condition with a larger tolerance δ. Steps 2 and 3 are constrained optimization problems, which may have local extrema. A global optimization algorithm is therefore recommended (Section 4.3.9). Examples of application are presented in (Pronzato and Walter, 1988).

Approximate design. When $\mathbb{P}$ is finite, the issue is to find a design measure m_{MM} that maximizes

$$j_{\mathrm{MM}}(m) = \min_{\mathbf{p} \in \{\mathbf{p}^1, \ldots, \mathbf{p}^m\}} j(\mathbf{p}, m).$$

This problem is closely connected to T-optimal design for discriminating among m model structures, $m > 2$ (Section 6.6.3.1). The fact that the cost function may be not differentiable at the optimum makes the determination of the optimal design measure difficult. Specific algorithms (for T-optimal design) are presented in (Atkinson and Fedorov, 1975b; Huang, 1991; Huang, Pronzato and Walter, 1991). The results described in (Wong, 1992) may be useful when $\mathbb{P}$ is not finite.

6.5 Design for Bayesian estimation

The Cramér-Rao inequality for an unbiased Bayesian estimator (Section 5.3.2) suggests replacing the average Fisher information matrix per sample by:

$$\mathbf{F}_{\mathrm{B}} = \frac{1}{n_t}\Big[\mathop{\mathrm{E}}_{\mathbf{p}}\{\mathbf{F}(\mathbf{p})\} + \mathop{\mathrm{E}}_{\mathbf{p}}\{[\frac{\partial}{\partial \mathbf{p}} \ln \pi_{\mathrm{p}}(\mathbf{p})][\frac{\partial}{\partial \mathbf{p}} \ln \pi_{\mathrm{p}}(\mathbf{p})]^{\mathrm{T}}\}\Big],$$

where $\mathbf{F}(\mathbf{p})$ is the Fisher information matrix and $\pi_{\mathrm{p}}(\mathbf{p})$ the prior density of the parameters. When the model structure is not LP, $\mathrm{E}_{\mathbf{p}}\{\mathbf{F}(\mathbf{p})\}$ cannot generally be calculated analytically. The optimization of any cost function $\phi_k(\mathbf{F}_{\mathrm{B}})$, with ϕ_k defined as in Section 6.1, then requires evaluation of the expected value of a matrix for each evaluation of the cost function. (Note that a stochastic approximation algorithm as in Section 6.4.3.2 cannot be used here.) For that reason, this approach seems to have been considered for LP-model structures only, *i.e.* when

$$\mathbf{y}^{\mathrm{s}}(\Xi) = \mathbf{R}(\Xi)\mathbf{p}^* + \mathbf{n}.$$

In particular, when the noise $\mathbf{n}$ has a Gaussian density $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Sigma}$ known, and the prior density is normal $\mathcal{N}(\mathbf{p}_0, \mathbf{\Omega})$, we have

$$\mathbf{F}_{\mathrm{B}}(\Xi) = \frac{1}{n_t}[\mathbf{R}^{\mathrm{T}}(\Xi)\mathbf{\Sigma}^{-1}\mathbf{R}(\Xi) + \mathbf{\Omega}^{-1}],$$

which is called the *Bayesian information matrix* (Pilz, 1983). If $\mathbf{\Sigma} = \sigma^2\mathbf{I}_{n_t}$, then

$$\mathbf{F}_{\mathrm{B}}(\Xi) = \frac{1}{\sigma^2 n_t}\left[\mathbf{R}^{\mathrm{T}}(\Xi)\mathbf{R}(\Xi) + \left(\frac{\mathbf{\Omega}}{\sigma^2}\right)^{-1}\right],$$

so $\mathbf{\Omega}$ and σ^2 only affect the determination of the optimal experiment through their ratio. In Section 5.3.2, we showed that the prior expectation of the minimum quadratic risk associated with a weighting matrix $\mathbf{K}^{\mathrm{T}}\mathbf{K}$ is

$$\mathop{\mathrm{E}}_{\mathbf{y}^{\mathrm{s}}}\{j_{\mathrm{mr}}(\hat{\mathbf{p}}_{\mathrm{map}}[\mathbf{y}^{\mathrm{s}}(\Xi)])\} = \mathrm{trace}\,\{\mathbf{K}[\mathbf{R}^{\mathrm{T}}(\Xi)\mathbf{\Sigma}^{-1}\mathbf{R}(\Xi) + \mathbf{\Omega}^{-1}]^{-1}\mathbf{K}^{\mathrm{T}}\},$$

which defines an L_B*-optimality* cost function

$$j_{\mathrm{LB}}(\Xi) = \mathrm{trace}\,[\mathbf{K}^{\mathrm{T}}\mathbf{K}\mathbf{F}_{\mathrm{B}}^{-1}(\Xi)].$$

More generally, for a non-LP model structure one can consider the cost function

$$j_{\mathrm{LB}}(\Xi) = \mathrm{trace}\,[\mathbf{Q}\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi)],$$

with

$$\mathbf{F}_{\mathrm{B}}(\mathbf{p}, \Xi) = \frac{1}{n_{\mathrm{t}}}\,[\mathbf{F}(\mathbf{p}, \Xi) + \boldsymbol{\Omega}^{-1}],$$

where

$$\mathbf{F}(\mathbf{p}, \Xi) = \sum_{i=1}^{n_{\mathrm{t}}} \frac{1}{w(\xi^i)}\,\mathbf{s}_{\mathrm{y}}(\xi^i, \mathbf{p})\mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi^i, \mathbf{p}).$$

6.5.1 Exact design

The $\mathrm{L_B}$-optimality cost function can be minimized by the following algorithm (Pilz, 1983), very similar to the DETMAX algorithm presented in Section 6.2.1.2.

Step 1: Choose Ξ^1 (with n_{t} support points) and set $k = 1$.
Step 2: Find

$$\xi^* = \arg\max_{\xi \in \xi} \frac{\mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi, \mathbf{p})\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^k)\mathbf{Q}\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^k)\mathbf{s}_{\mathrm{y}}(\xi, \mathbf{p})}{n_{\mathrm{t}}w(\xi) + \mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi, \mathbf{p})\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^k)\mathbf{s}_{\mathrm{y}}(\xi, \mathbf{p})}.$$

Update $\mathbf{F}_{\mathrm{B}}^{-1}$ for the experiment $\Xi^{k+} = (\Xi^{k\mathrm{T}}, \xi^{*\mathrm{T}})^{\mathrm{T}}$ according to

$$\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^{k+}) = \frac{n_{\mathrm{t}}+1}{n_{\mathrm{t}}}\left[\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^k) - \frac{\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^k)\mathbf{s}_{\mathrm{y}}(\xi^*, \mathbf{p})\mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi^*, \mathbf{p})\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^k)}{n_{\mathrm{t}}w(\xi^*) + \mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi^*, \mathbf{p})\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^k)\mathbf{s}_{\mathrm{y}}(\xi^*, \mathbf{p})}\right].$$

Step 3: Find

$$\xi^{i*} = \arg\min_{\xi^i \in \operatorname{supp}[\Xi^{k+}]} \frac{\mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi^i, \mathbf{p})\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^{k+})\mathbf{Q}\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^{k+})\mathbf{s}_{\mathrm{y}}(\xi^i, \mathbf{p})}{(n_{\mathrm{t}}+1)w(\xi^i) - \mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi^i, \mathbf{p})\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^{k+})\mathbf{s}_{\mathrm{y}}(\xi^i, \mathbf{p})}.$$

Step 4: If $\xi^{i*} = \xi^*$, stop. Else remove ξ^{i*} from Ξ^{k+} to get Ξ^{k+1} and update $\mathbf{F}_{\mathrm{B}}^{-1}$ according to

$$\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^{k+1}) =$$

$$\frac{n_{\mathrm{t}}}{n_{\mathrm{t}}+1}\left[\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^{k+}) + \frac{\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^{k+})\mathbf{s}_{\mathrm{y}}(\xi^{i*}, \mathbf{p})\mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi^{i*}, \mathbf{p})\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^{k+})}{(n_{\mathrm{t}}+1)w(\xi^{i*}) - \mathbf{s}_{\mathrm{y}}^{\mathrm{T}}(\xi^{i*}, \mathbf{p})\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^{k+})\mathbf{s}_{\mathrm{y}}(\xi^{i*}, \mathbf{p})}\right].$$

Increment k by one, and go to Step 2.

Note that $\mathbf{F}_{\mathrm{B}}(\mathbf{p}, \Xi)$ is positive-definite for any non-negative-definite matrix $\mathbf{F}(\mathbf{p}, \Xi)$, provided $\boldsymbol{\Omega}$ is positive-definite. It therefore does not matter if the initial experiment is degenerate. A possible choice for Ξ^1 is obtained as follows.

Step 1.1: Set $n = 1$, $\Xi^1 = \{\varnothing\}$ and $\mathbf{F}_{\mathrm{B}}^{-1}(\mathbf{p}, \Xi^1) = \boldsymbol{\Omega}$.
Step 1.2: Perform Step 2 with n_{t} replaced by n.
Step 1.3: If $n = n_{\mathrm{t}}$ stop. Else increment n by one and go to Step 1.2.

As with the exact-design algorithms of Section 6.2.1, convergence to an L_B-optimal experiment is not guaranteed. The choice of the initial experiment Ξ^1 is thus especially critical, and it is recommended that the optimization be repeated, initialized at different experiments. As for the DETMAX algorithm, excursions with length λ greater than one can be considered. Similar algorithms may be used (Pilz, 1983) to construct a *D_B-optimal experiment*, which maximizes

$$j_{DB}(\Xi) = \det \mathbf{F}_B(\mathbf{p}, \Xi).$$

6.5.2 Approximate design

The approximate design theory presented in Section 6.2.2 also applies to Bayesian estimators. Consider the matrix

$$\mathbf{F}_B(\mathbf{p}, m) = \int_{\xi} \frac{1}{w(\xi)} \mathbf{s}_y(\xi, \mathbf{p})\mathbf{s}_y^T(\xi, \mathbf{p})m(d\xi) + \frac{\mathbf{\Omega}^{-1}}{n_t} = \mathbf{F}_{ps}(\mathbf{p}, m) + \frac{\mathbf{\Omega}^{-1}}{n_t} .$$

(Note that the value of n_t must be specified.) Caratheodory's theorem can again be used to show that $[n_p(n_p + 1)/2] + 1$ support points are enough to construct any matrix $\mathbf{F}_B(\mathbf{p}, m)$. In the particular case of L_B-optimality, one can show that $\rho(2n_p - \rho - 1)/2$ support points are enough, with $\rho = \text{rank } \mathbf{Q}$ (Pilz, 1983; Chaloner, 1984). One can also show that the matrix $\mathbf{F}_B(\mathbf{p}, m)$ associated with an L_B-optimal design measure, with $\mathbf{Q} = \mathbf{I}_{n_p}$, or with a D_B-optimal design measure, is unique. Equivalence theorems, see Section 6.2.2.3, can be proved for cost functions with suitable convexity (or concavity) properties.

EQUIVALENCE THEOREMS (Pilz, 1983; Chaloner, 1984)

— The design measure m_{LB} is L_B-optimal if and only if

$$\max_{\xi \in \xi} \frac{1}{w(\xi)} \mathbf{s}_y^T(\xi, \mathbf{p})\mathbf{F}_B^{-1}(\mathbf{p}, m_{LB})\mathbf{Q}\mathbf{F}_B^{-1}(\mathbf{p}, m_{LB})\mathbf{s}_y(\xi, \mathbf{p}) \leq$$
$$\text{trace } [\mathbf{F}_B^{-1}(\mathbf{p}, m_{LB})\mathbf{Q}\mathbf{F}_B^{-1}(\mathbf{p}, m_{LB})\mathbf{F}_m(\mathbf{p}, m_{LB})].$$

— The design measure m_{DB} is D_B-optimal if and only if

$$\max_{\xi \in \xi} \frac{1}{w(\xi)} \mathbf{s}_y^T(\xi, \mathbf{p})\mathbf{F}_B^{-1}(\mathbf{p}, m_{DB})\mathbf{s}_y(\xi, \mathbf{p}) \leq \text{trace } [\mathbf{F}_B^{-1}(\mathbf{p}, m_{DB})\mathbf{F}_m(\mathbf{p}, m_{DB})]. \qquad \Diamond$$

Optimization algorithms similar to that of Fedorov and Wynn presented in Section 6.2.2.4 can be constructed from this theorem. Their convergence to the global optimum is guaranteed. The function $d(m, \xi)$ is simply replaced by

$$d_{LB}(m, \xi) = \mathbf{s}_y^T(\xi, \mathbf{p})\mathbf{F}_B^{-1}(\mathbf{p}, m)\mathbf{Q}\mathbf{F}_B^{-1}(\mathbf{p}, m)\mathbf{s}_y(\xi, \mathbf{p})/w(\xi)$$

for L_B-optimality, and by

$$d_{DB}(m, \xi) = \mathbf{s}_y^T(\xi, \mathbf{p})\mathbf{F}_B^{-1}(\mathbf{p}, m)\mathbf{s}_y^T(\xi, \mathbf{p})/w(\xi)$$

for D_B-optimality. The stopping rule is obtained from the necessary and sufficient optimality condition of the corresponding equivalence theorem. The initial measure m^1 is allowed to be degenerate. In particular, the measure with a single support point

$$\begin{aligned}\xi^1 &= \arg \min_{\xi \in \Xi} \operatorname{trace}\left\{\mathbf{Q}\left[\frac{1}{w(\xi)}\mathbf{s}_y(\xi, \mathbf{p})\mathbf{s}_y^T(\xi, \mathbf{p}) + \frac{\boldsymbol{\Omega}^{-1}}{n_t}\right]^{-1}\right\} \\ &= \arg \max_{\xi \in \Xi} \frac{\mathbf{s}_y^T(\xi, \mathbf{p})\boldsymbol{\Omega}\mathbf{Q}\boldsymbol{\Omega}\mathbf{s}_y(\xi, \mathbf{p})}{w(\xi) + n_t\mathbf{s}_y^T(\xi, \mathbf{p})\boldsymbol{\Omega}\mathbf{s}_y(\xi, \mathbf{p})}\end{aligned}$$

can be used for L_B-optimality, and the measure

$$\begin{aligned}\xi^1 &= \arg \max_{\xi \in \Xi} \det\left[\frac{1}{w(\xi)}\mathbf{s}_y(\xi, \mathbf{p})\mathbf{s}_y^T(\xi, \mathbf{p}) + \frac{\boldsymbol{\Omega}^{-1}}{n_t}\right] \\ &= \arg \max_{\xi \in \Xi} \frac{1}{w(\xi)}\mathbf{s}_y^T(\xi, \mathbf{p})\boldsymbol{\Omega}\mathbf{s}_y(\xi, \mathbf{p})\end{aligned}$$

for D_B-optimality.

6.6 Influence of model structure

This problem is extremely important, although it has not had much attention, in particular for applications. As indicated in Section 1.1, the model may serve several objectives. Its identification will be considered successful only if the model suitably reproduces the pertinent aspects of the behaviour of the system, in the operating range of interest.

The structural properties of models, the choice of an estimator for the parameters of a model with given structure, and the numerical methods used for the computation of the estimates have been discussed in previous chapters. In choosing the structure, different situations must be distinguished. Assume that the data have already been collected.

— If the model structure is fixed, one should check its compatibility with the data, and some basic tools for that purpose will be indicated in Chapter 7.

— If a finite number of structures, for instance with increasing complexity, compete for the description of the data, the AIC criterion (Section 3.4), or any other criterion of the same type, can be used to select one of them. The validity of the structure chosen will then need to be tested.

— If the structure is completely unknown, one can sometimes use a simple parametric structure (*e.g.*, LP), for a very coarse description of the behaviour of the system, complemented by a nonparametric structure; see, *e.g.*, the kriging method of Section 3.3.5. Other approaches will be presented in Section 6.6.2.

A key point is that attempts to falsify a given structure or to select one structure from several candidates will be meaningful only if the data collected are rich enough. For instance, if maximum-likelihood estimation is used, one needs at least as many separate observations (*e.g.* at different times) as there are parameters in the most complex structure. For that reason, the problem of structure selection cannot easily be dissociated from that of experiment design.

The first approach to be considered merely aims, through Bayesian estimation, to design an experiment for a simple structure while being protected against the possible presence of neglected terms in the process response. Next, the statistical literature on robustness of estimation and design in the face of modelling errors will be briefly reviewed. Finally, we shall investigate an alternative approach, which consists of designing an experiment that allows us to choose the best model structure from a finite family. The experimental conditions must then be chosen to maximize the sensitivity of the response to modelling errors; this is a quantitative extension of the notion of structural distinguishability (Section 2.6.2).

6.6.1 Robustness through Bayesian estimation

As already indicated, a classical optimal experiment (D-optimal for instance) often consists of repeating observations under a small number of distinct experimental conditions (*e.g.* measurement times), sometimes equal to the number of parameters to be estimated. It is then impossible to estimate the larger number of parameters of a more complex structure from such an experiment. If hesitating between a simple and a complex structure, one should thus design the experiment for the complex one. However, the complex structure may be considered only as a precaution, and the experiment designed for it may prove far from optimal for the simple structure. A Bayesian approach may permit to bypass this difficulty, as illustrated by the following example.

EXAMPLE 6.13

Consider the model structures

$$y_{\mathrm{m}1}(\xi, \mathbf{p}) = p_1 + p_2\xi \quad \text{and} \quad y_{\mathrm{m}2}(\xi, \mathbf{p}) = p_1 + p_2\xi + p_3\xi^2,$$

with $\xi \in [-1, 1]$. Assume that the measurement errors are additive and correspond to i.i.d. $\mathcal{N}(0, \sigma^2)$ variables. The D-optimal design measure for $y_{\mathrm{m}1}$ is then

$$m_{\mathrm{D}1} = \left\{ \begin{array}{cc} -1 & 1 \\ 1/2 & 1/2 \end{array} \right\},$$

whereas for $y_{\mathrm{m}2}$ it is

$$m_{\mathrm{D}2} = \left\{ \begin{array}{ccc} -1 & 0 & 1 \\ 1/3 & 1/3 & 1/3 \end{array} \right\}.$$

m_{D1} does not allow all parameters of y_{m2} to be estimated, and m_{D2} has a D-efficiency of only 2/3 for estimation of the parameters of y_{m1}. The prior confidence in the model y_{m2} can be expressed as a prior density for its parameters (the more likely the first structure is, the more concentrated the marginal density of p_3 will be around $p_3 = 0$). Assume that $\mathbf{p} = (p_1, p_2, p_3)^T$ has a prior density $\mathcal{N}(\mathbf{p}_0, \mathbf{\Omega})$, with $\mathbf{\Omega} = \text{diag}\{\omega_1, \omega_2, \omega_3\}$. It is easy to check that the D_B-optimal design measure (Section 6.5) tends to m_{D1} as ω_3 tends to 0 (another way of saying that when p_3 is known it need not be estimated!). ◊

6.6.2 Robust estimation and design

In contrast to Section 3.7, estimation is understood here to be robust with respect to errors in the model structure, or more precisely neglected terms in the response (*e.g.*, when searching for the maximum of a measured response; see Section 4.4).

The response of the process is written as

$$y(\xi) = f(\xi) + \varepsilon(\xi),$$

with $f(\xi)$ the deterministic part and the $\varepsilon(\xi)$'s assumed to be i.i.d. $\mathcal{N}(0, \sigma^2)$. The influence of robustness on design seems first to have been considered by Box and Draper (1959), who studied the consequences of using a simple model

$$y_m(\xi, \mathbf{p}) = \mathbf{r}^T(\xi)\mathbf{p}$$

(for instance an affine function of ξ) when the true structure $f(\xi) = \mathbf{z}^T(\xi)\mathbf{q}$ is more complex (for instance a quadratic function of ξ). The error in the structure then produces a bias in the estimated parameters, and the integral of the mean-square error (called the *J-criterion* in the literature)

$$J(\hat{\mathbf{p}}, \Xi, f) = \underset{\mathbf{y}^s|\mathbf{q},\Xi}{\mathrm{E}} \left\{ \int_{\xi} [f(\xi) - \mathbf{r}^T(\xi)\hat{\mathbf{p}}(\mathbf{y}^s)]^2 \, d\xi \right\}$$

is the sum of a bias term and a variance term. Box and Draper use the least-squares estimator $\hat{\mathbf{p}}_{ls}(\mathbf{y}^s)$ and consider only the bias term in choosing the experiment. Karson, Manson and Hader (1969) use a linear estimator and choose the experiment that minimizes the variance term under the constraint that the bias term is minimal. These two approaches are for exact design. Kiefer (1973) considers minimization of $J(\hat{\mathbf{p}}, \Xi, f)$ for approximate design when a linear estimator of $\mathbf{p}$ is used. He shows that the cost function corresponds to L_B-optimality (Section 6.5) when a prior distribution is used for the parameters $\mathbf{q}$. Stigler (1971) and Studden (1982) study D-optimal design for models polynomial in ξ with the constraint that the parameters of a higher-degree model can be estimated with a guaranteed degree of precision.

In most practical situations, the structure of the deterministic part $f(\xi)$ of the process response is partially unknown, and two classes of methods have been proposed to deal with this lack of information.

6.6.2.1 Minimax approach

The robust-design approaches above might not guard against small deviations from the assumed model when these deviations do not correspond to the terms (most often polynomials) arbitrarily selected. This is why Huber (1975) suggests minimizing the supremum of the integrated mean-square error

$$J(\hat{\mathbf{p}}, \Xi, f) = \underset{\mathbf{y}^s|f,\Xi}{\mathrm{E}} \left\{ \int_{\xi} [f(\xi) - y_m[\xi, \hat{\mathbf{p}}(\mathbf{y}^s)]]^2 \, d\xi \right\},$$

with respect to f, where f belongs to a given (infinite dimensional) family of functions. Although this approach is theoretically appealing, the results obtained only deal with the one-dimensional regression model $y_m(\xi, \mathbf{p}) = p_0 + p_1\xi$, with a linear estimator for $\mathbf{p} = (p_1, p_2)^T$. Moreover, Markus and Sacks (1977) have stressed that the design measures obtained with this approach must be absolutely continuous with respect to the Lebesgue measure and are therefore generally not implementable. They suggest, for the same model structure, minimization of

$$j(\hat{\mathbf{p}}, m) = \sup_{f} \underset{\mathbf{y}^s|f,m}{\mathrm{E}} \{[\hat{p}_0(\mathbf{y}^s) - p_0^*]^2 + \alpha^2[\hat{p}_1(\mathbf{y}^s) - p_1^*]^2\},$$

where α is fixed and $f(\xi) = p_0^* + p_1^*\xi + \omega(\xi)$, with $|\omega(\xi)| \leq m(\xi)$ and m a known function of ξ.

A more general situation, is when $y_m(\xi, \mathbf{p}) = \mathbf{r}^T(\xi)\mathbf{p}$, with

$$f(\xi) = y_m(\xi, \mathbf{p}) + \omega(\xi),$$

where $|\omega(\xi)| \leq m(\xi)$, with m a known function of ξ. This case is considered in the rest of this section. The linear estimator of a linear function of $\mathbf{p}$ (*e.g.*, a predictor of the process output) minimizing the supremum of a quadratic error with respect to f is constructed in (Sacks and Ylvisaker, 1978). However, estimation of $\mathbf{p}$ (or prediction of several linear combinations of $\mathbf{p}$) requires solution of a difficult optimization problem. Mathew and Nordström (1993) consider instead the minimax cost function

$$j(\mathbf{p}) = \sup_{f} [\mathbf{y}^s - \mathbf{R}(\Xi)\mathbf{p} - \boldsymbol{\omega}(\Xi)]^T \mathbf{W} [\mathbf{y}^s - \mathbf{R}(\Xi)\mathbf{p} - \boldsymbol{\omega}(\Xi)],$$

with $\Xi = (\xi^{1T}, \ldots, \xi^{mT})^T$, $\boldsymbol{\omega}(\Xi) = [\omega(\xi^1), \ldots, \omega(\xi^m)]^T$ and $\mathbf{W}$ a diagonal weighting matrix. They show that this cost function can be written as a linear combination of L_2 (least-squares) and L_1 (least-modulus) cost functions, and that it is convex with respect to $\mathbf{p}$, which facilitates its optimization.

The general design problem for estimation of a linear functional of a regression model belonging to a given class of functions is considered in (Sacks and Ylvisaker, 1984). Pesotchinsky (1982) defines a mean-square error matrix for $\hat{\mathbf{p}}_{ls}$ (for an LP model structure linear with respect to ξ) and constructs minimax extensions for design cost functions in the ϕ_k family; see (Kiefer, 1974) and Section 6.1. Welch (1983) considers a minimax cost function based on the integral of the mean-square error

$$j(\Xi) = \sup_{f} \mathop{\mathrm{E}}_{\mathbf{y}^s|f,\Xi} \{ \int_{\xi} [f(\xi) - \mathbf{r}^{\mathrm{T}}(\xi)\hat{\mathbf{p}}_{\mathrm{ls}}(\mathbf{y}^s)]^2 \, d\xi \},$$

with $f(\xi) = \mathbf{r}^{\mathrm{T}}(\xi)\mathbf{p} + \omega(\xi)$, and $|\omega(\xi)| \leq \omega_{\max}$. He suggests algorithms similar to the DETMAX algorithm for exact design (Section 6.2.1.2) or to the Fedorov-Wynn algorithm for approximate design (Section 6.2.2.4).

6.6.2.2 Bayesian approach

An alternative approach to account for prior uncertainty in the model structure relies on a Bayesian statistical model. O'Hagan (1978) introduces the notion of a localized regression model, for which $f(\xi) = \mathbf{r}^{\mathrm{T}}(\xi)\mathbf{p}(\xi)$ (with ξ scalar). Keeping $\mathbf{p}$ constant, one gets an LP structure approximating f, whereas a deviation from the LP restriction is obtained by letting $\mathbf{p}$ vary with ξ, and assuming that the correlation between $\mathbf{p}(\xi^1)$ and $\mathbf{p}(\xi^2)$ increases as $|\xi^1 - \xi^2|$ falls. Knowledge about $\mathbf{p}(\xi)$ is then introduced through a prior density such that

$$\mathrm{E}\{\mathbf{p}(\xi)\} = \mathbf{p}_0 \text{ for all } \xi, \text{ and } \mathrm{E}\{[\mathbf{p}(\xi^1) - \mathbf{p}_0][\mathbf{p}(\xi^2) - \mathbf{p}_0]^{\mathrm{T}}\} = \rho(|\xi^1 - \xi^2|)\mathbf{M},$$

with $\mathbf{p}_0$ and $\mathbf{M}$ fixed, and ρ a monotononically decreasing function in $\mathbb{R}^+$ with $\rho(0) = 1$. The joint density of the $\mathbf{p}(\xi)$'s is assumed to be normal. The posterior mean $\mathrm{E}_{\mathbf{p}(\xi)|\mathbf{y}^s}\{\mathbf{p}(\xi)\}$ is then obtained analytically, and can be used to predict the response at any ξ, through

$$\hat{y}(\xi) = \mathbf{r}^{\mathrm{T}}(\xi) \mathop{\mathrm{E}}_{\mathbf{p}(\xi)|\mathbf{y}^s} \{\mathbf{p}(\xi)\}.$$

The proposed design criterion is based on the mean-square prediction error. The flexibility of the approach allows, in principle, realistic modelling in a variety of situations. A serious difficulty, however, is the choice of values for the prior parameters; see the discussion of (O'Hagan, 1978). Strong similarities exist with the linear Bayes regression estimator of Goldstein (1980) and the Bayesian model used by Steinberg (1985) for response-surface problems.

Finally, kriging (Section 3.3.5) can also be considered as an attempt to achieve robustness with respect to the model structure. It describes $\omega(\xi) = f(\xi) - y_{\mathrm{m}}(\xi, \mathbf{p})$ as a realization of a Gaussian process with known (or estimated) statistics. A Bayesian formulation is presented in (Blight and Ott, 1975; Currin *et al.*, 1991).

6.6.3 Structure discrimination

Consider now the problem of determining experimental conditions that will allow the best possible discrimination between competing model structures. The literature devoted to this problem is almost as rich as that on experiment design for parameter estimation. We restrict ourselves to presenting the main ideas, and recommend the survey papers of Atkinson and Cox (1974), and Hill (1978). Much of this section is derived from (Huang, 1991). Three approaches will be considered, based respectively on the notions of prediction discrepancy, entropy and $\mathrm{D_s}$-optimality. To simplify the presentation, the additive measurement errors will be assumed to be scalar and i.i.d. $\mathcal{N}(0, \sigma^2)$.

6.6.3.1 Discriminating by prediction discrepancy

The use of prediction discrepancy, initially suggested by Fedorov and Pázman (1968), resulted in the theoretical notion of *T-optimality* (where T stands for Testing), related to the power of a χ^2 test (Atkinson and Fedorov, 1975a). Consider first the case of two rival structures.

Discriminating between two structures. Assume first that the true structure is known and indexed by one, with $\mathbf{p}_1^*$ the true value of its parameters. We shall see later how this hypothesis can be relaxed. The observations thus correspond to

$$y(\xi^i) = y_{\mathrm{m}_1}(\xi^i, \mathbf{p}_1^*) + \varepsilon(\xi^i), \quad i = 1, \dots, n_t.$$

The cost function for an experiment $\Xi = (\xi^{1\mathrm{T}}, \dots, \xi^{n_t\mathrm{T}})^{\mathrm{T}}$ is

$$\Delta_2^{(1)}(\Xi) = \sum_{i=1}^{n_t} \left\{ y_{\mathrm{m}_1}(\xi^i, \mathbf{p}_1^*) - y_{\mathrm{m}_2}[\xi^i, \hat{\mathbf{p}}_2(\Xi)] \right\}^2,$$

where $\hat{\mathbf{p}}_2(\Xi)$ is the least-squares estimator of the parameters $\mathbf{p}_2 \in \mathbb{P}_2$ of the second structure, obtained from the fictitious (noise-free) data $y_{\mathrm{m}_1}(\xi^i, \mathbf{p}_1^*)$ $(i = 1, \dots, n_t)$, *i.e.*

$$\hat{\mathbf{p}}_2(\Xi) = \arg \min_{\mathbf{p}_2 \in \mathbb{P}_2} \sum_{i=1}^{n_t} \left[y_{\mathrm{m}_1}(\xi^i, \mathbf{p}_1^*) - y_{\mathrm{m}_2}(\xi^i, \mathbf{p}_2) \right]^2.$$

The cost function $\Delta_2^{(1)}(\Xi)$ is called *non-centrality parameter* of the structure $y_{\mathrm{m}2}$ (Atkinson and Fedorov, 1975a). When approximate design theory is used, with m a normalized design measure on the set $\mathbb{\xi}$ of admissible experimental conditions (Section 6.2.2), the cost function becomes

$$\Delta_2^{(1)}(m) = \min_{\mathbf{p}_2 \in \mathbb{P}_2} \int_{\mathbb{\xi}} \left[y_{\mathrm{m}_1}(\xi, \mathbf{p}_1^*) - y_{\mathrm{m}_2}(\xi, \mathbf{p}_2) \right]^2 m(\mathrm{d}\xi),$$

which is concave with respect to m. A design measure m_{T} is said to be T-optimal if it maximizes $\Delta_2^{(1)}(m)$. Atkinson and Fedorov (1975a) have proved the following result, which summarizes the properties of T-optimal design measures.

EQUIVALENCE THEOREM

— A necessary and sufficient condition for m_{T} to be T-optimal is

$$\{ y_{\mathrm{m}_1}(\xi, \mathbf{p}_1^*) - y_{\mathrm{m}_2}[\xi, \hat{\mathbf{p}}_2(m_{\mathrm{T}})] \}^2 \leq \Delta_2^{(1)}(m_{\mathrm{T}})$$

for all ξ in $\mathbb{\xi}$.

— $\{y_{\mathrm{m}_1}(\xi, \mathbf{p}_1^*) - y_{\mathrm{m}_2}[\xi, \hat{\mathbf{p}}_2(m_{\mathrm{T}})]\}^2$ reaches its upper bound when ξ is a support point of m_{T}.
— The set of T-optimal design measures is convex. ◊

From this theorem an optimization algorithm can be derived, similar to the Fedorov-Wynn algorithm of Section 6.2.2.4, but with $d(\xi, m)$ replaced by $\{y_{\mathrm{m}_1}(\xi, \mathbf{p}_1^*) - y_{\mathrm{m}_2}[\xi, \hat{\mathbf{p}}_2(m)]\}^2$. Its global convergence is guaranteed. However, the assumption of a known true structure with known parameters is not realistic, and a sequential approach must be used in practice. The algorithm is then as follows (Atkinson and Fedorov, 1975a):

Step 1: After k observations (with the experimental conditions Ξ^k), estimate the parameters $\hat{\mathbf{p}}_1(\Xi^k)$ and $\hat{\mathbf{p}}_2(\Xi^k)$ of both structures (in the least-squares sense).
Step 2: Choose the (k+1)th experimental conditions (support point) ξ^{k+1} such that

$$\xi^{k+1} = \arg\max_{\xi\in\xi} \left\{ y_{\mathrm{m}_1}[\xi, \hat{\mathbf{p}}_1(\Xi^k)] - y_{\mathrm{m}_2}[\xi, \hat{\mathbf{p}}_2(\Xi^k)] \right\}^2.$$

Step 3: Increment k by one and go to Step 1.

When the algorithm converges to a design measure that is non-degenerate for both structures, this design measure is almost surely T-optimal (Fedorov, 1975).

Discriminating between more than two structures. The notion of T-optimality can be extended to m ($m > 2$) rival structures (Atkinson and Fedorov, 1975b). Assume again that the true structure is y_{m_1} with true parameters $\mathbf{p}_1^*$. The cost function then becomes

$$\Delta^{(1)}(m) = \min_{i\neq 1} \Delta_i^{(1)}(m).$$

The possible non-differentiability of $\Delta^{(1)}$ at the optimum makes the design of a T-optimal experiment difficult (Atkinson and Fedorov, 1975b; Huang, 1991). A possible sequential policy is as follows:

Step 1: After k observations with experimental conditions Ξ^k, rank the m sums of squares of residuals,

$$\rho_{i_1}^k \leq \rho_{i_2}^k \leq \ldots \leq \rho_{i_m}^k,$$

and construct the set $\mathbb{I}_1^{\nu}(\Xi^k)$ defined by

$$\mathbb{I}_1^{\nu}(\Xi^k) = \{ i \in \{1, \ldots, m\} \mid \frac{\rho_i^k - \rho_{i_2}^k}{k} \leq \nu,\ i \neq i_1 \}$$

(which contains i_2 whatever the tolerance $\nu > 0$).
Step 2: If $\mathbb{I}_1^{\nu}(\Xi^k)$ is a singleton, $\mathbb{I}_1^{\nu}(\Xi^k) = \{i_2\}$, take

$$\boldsymbol{\xi}^{k+1} = \arg \max_{\boldsymbol{\xi} \in \mathbb{\xi}} \left\{ y_{\mathrm{m}i_1}[\boldsymbol{\xi}, \hat{\mathbf{p}}_{i_1}(\Xi^k)] - y_{\mathrm{m}i_2}[\boldsymbol{\xi}, \hat{\mathbf{p}}_{i_2}(\Xi^k)] \right\}^2,$$

else solve the minimax problem

$$(\boldsymbol{\alpha}^*, \boldsymbol{\xi}^{k+1}) = \arg \min_{\boldsymbol{\alpha} \in \mathbb{A}} \max_{\boldsymbol{\xi} \in \mathbb{\xi}} \sum_{i \in \mathbb{I}_1^{\nu}(\Xi^k)} \alpha_i \left\{ y_{\mathrm{m}i_1}[\boldsymbol{\xi}, \hat{\mathbf{p}}_{i_1}(\Xi^k)] - y_{\mathrm{m}i}[\boldsymbol{\xi}, \hat{\mathbf{p}}_{i}(\Xi^k)] \right\}^2,$$

with

$$\mathbb{A} = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^m \mid \alpha_i = 0 \text{ if } i \notin \mathbb{I}_1^{\nu}(\Xi^k),\ \alpha_i \geq 0 \text{ if } i \in \mathbb{I}_1^{\nu}(\Xi^k), \quad \sum_{i \in \mathbb{I}_1^{\nu}(\Xi^k)} \alpha_i = 1 \right\}.$$

Step 3: Increment k by one and go to Step 1.

The tolerance ν can be taken as decreasing with k, but less rapidly than $1/\sqrt{k}$ (Atkinson and Fedorov, 1975b). The minimax problem at Step 2 may be solved by a relaxation algorithm similar to that in Section 6.4.4.2.

6.6.3.2 Discriminating *via* entropy

Let $\pi_1^k, \ldots, \pi_m^k$ be the probabilities associated with the m rival structures after k observations. The corresponding Shannon entropy is

$$\mathcal{H}_m^k = - \sum_{i=1}^{m} \pi_i^k \ln \pi_i^k.$$

It is a maximum when all structures have the same probability ($\pi_i^k = 1/m$, $i = 1, \ldots, m$). If a structure has probability one of being correct, the entropy takes its minimal value zero, hence the intuitive idea of minimizing entropy, or, equivalently, of maximizing the decrease of entropy $\Delta\mathcal{H}_m^{k+1}$ due to the (k+1)th observation,

$$\Delta\mathcal{H}_m^{k+1} = \mathcal{H}_m^k + \sum_{n=1}^{m} \pi_n^{k+1}[y(k+1)] \ln \pi_n^{k+1}[y(k+1)],$$

with

$$\pi_n^{k+1}[y(k+1)] = \frac{\pi_n^k \, \pi_n[y(k+1)|\mathbf{y}_1^k]}{\sum_{i=1}^{m} \pi_i^k \, \pi_i[y(k+1)|\mathbf{y}_1^k]},$$

where $\pi_n(y|\mathbf{y}_1^k)$ denotes the probability density of y for the nth structure after the k observations $\mathbf{y}_1^k = [y(1), \ldots, y(k)]^{\mathrm{T}}$, with experimental conditions Ξ^k. If y is obtained with experimental conditions $\boldsymbol{\xi}$, one has

$$\pi_n(y|\mathbf{y}_1^k) = \frac{1}{\sqrt{2\pi[\sigma^2 + \sigma_n^2(\xi)]}} \exp\left(-\frac{1}{2}\frac{\{y - y_{\mathrm{m}_n}[\xi, \hat{\mathbf{p}}_n(\Xi^k)]\}^2}{\sigma^2 + \sigma_n^2(\xi)}\right),$$

with $\sigma_n^2(\xi)$ the (approximated) variance of the prediction of the response of the nth structure at the support point ξ, given by

$$\sigma_n^2(\xi) = \mathbf{s}^{\mathrm{T}}_{\mathrm{y}_n}[\xi, \hat{\mathbf{p}}_n(\Xi^k)]\ \mathbf{F}_n^{-1}[\hat{\mathbf{p}}_n(\Xi^k), \Xi^k]\ \mathbf{s}_{\mathrm{y}_n}[\xi, \hat{\mathbf{p}}_n(\Xi^k)],$$

where $\mathbf{s}_{\mathrm{y}_n}[\xi, \hat{\mathbf{p}}_n(\Xi^k)]$ and $\mathbf{F}_n[\hat{\mathbf{p}}_n(\Xi^k), \Xi^k]$ are respectively the sensitivity of the model response and the Fisher information matrix for the nth structure at the estimate $\hat{\mathbf{p}}_n(\Xi^k)$ of its parameters, obtained after the k observations under experimental conditions Ξ^k. This expression is obtained by linearization of the model response at $\hat{\mathbf{p}}_n(\Xi^k)$.

Δh_m^{k+1} depends on $y(k+1)$ which is unknown, so we consider instead its prior expectation,

$$\mathrm{E}\{\Delta h_m^{k+1}\} = \int \Delta h_m^{k+1} \pi_{\mathrm{y}}[y(k+1)|\mathbf{y}_1^k]\mathrm{d}y(k+1),$$

with

$$\pi_{\mathrm{y}}[y(k+1)|\mathbf{y}_1^k] = \sum_{n=1}^{m} \pi_n^k\ \pi_n[y(k+1)|\mathbf{y}_1^k].$$

$\mathrm{E}\{\Delta h_m^{k+1}\}$ cannot be obtained analytically, and Box and Hill (1967) suggest replacing it by an upper bound $\delta \geq \mathrm{E}\{\Delta h_m^{k+1}\}$, which can. This leads to the following cost function (to be maximized)

$$\delta(\xi) = \frac{1}{2}\sum_{i=1}^{m}\ \sum_{n=i+1}^{m} \pi_i^k\ \pi_n^k \Bigg(\frac{[\sigma_i^2(\xi) - \sigma_n^2(\xi)]^2}{[\sigma^2 + \sigma_i^2(\xi)][\sigma^2 + \sigma_n^2(\xi)]} +$$
$$\{y_{\mathrm{m}_i}[\xi, \hat{\mathbf{p}}_i(\Xi^k)] - y_{\mathrm{m}_n}[\xi, \hat{\mathbf{p}}_n(\Xi^k)]\}^2\left[\frac{1}{\sigma^2 + \sigma_i^2(\xi)} + \frac{1}{\sigma^2 + \sigma_n^2(\xi)}\right]\Bigg),$$

The $(k+1)$th observation is then taken at the support point ξ^{k+1} that maximizes $\delta(\xi)$, and the probabilities of the different structures are updated according to the expression above for $\pi_n^{k+1}[y(k+1)]$. (Atkinson (1978) shows that, for structures with different complexities, this approach tends to favour those with fewer parameters.) If there is no clear indication that one of the structures should be preferred, the parameters of each are estimated and the procedure is iterated.

Compared to that based on prediction discrepancy, this approach based on entropy has the advantages of having an intuitively appealing stopping criterion and of not requiring solution of a minimax problem when there are more than two rival structures. Reilly (1970) suggests replacing the calculation of a bound on the expected variation of entropy by its numerical evaluation (a modification of the Gauss-Hermite integration method), which does not seem to yield significantly different results (Huang, 1991). Note that if only two structures are considered, their probabilities do not take part in the determination of ξ^{k+1}.

These approaches based on prediction discrepancy and entropy are both sequential, and a comparison on various simulated examples yielded closely similar experiments

(Huang, 1991). The approach presented in the next section permits consideration of non-sequential discrimination problems for LP structures.

6.6.3.3 Discriminating *via* D$_s$-optimality

Consider first the simplest case where there are only two LP rival structures,

$$y_{mk}(\xi, \mathbf{p}_k) = \mathbf{r}_k^T(\xi)\mathbf{p}_k, \text{ with } \dim \mathbf{p}_k = n_k, \quad k = 1, 2.$$

If the true structure is the first, with parameters $\mathbf{p}_1$, a possible cost function is the non-centrality parameter, already considered in Section 6.6.3.1:

$$\Delta_2^{(1)}(m) = \min_{\mathbf{p}_2 \in \mathbb{P}_2} \int_{\xi} [y_{m1}(\xi, \mathbf{p}_1) - y_{m2}(\xi, \mathbf{p}_2)]^2 m(\mathrm{d}\xi),$$

which, when $\mathbb{P}_2 = \mathbb{R}^{n_2}$, gives

$$\Delta_2^{(1)}(m) = \mathbf{p}_1^T \mathbf{F}_2^{(1)}(m)\mathbf{p}_1,$$

with

$$\mathbf{F}_2^{(1)}(m) = \mathbf{F}_{11}(m) - \mathbf{F}_{12}(m)\mathbf{F}_{22}^{-1}(m)\mathbf{F}_{21}(m),$$

and

$$\mathbf{F}_{ik}(m) = \int_{\xi} \mathbf{r}_i(\xi)\mathbf{r}_k^T(\xi) m(\mathrm{d}\xi), \quad i, k = 1, 2.$$

REMARK 6.13

When the two structures have a common part, it may be removed from the parameter vector $\mathbf{p}_1$ and regressor vector $\mathbf{r}_1$ to write $\Delta_2^{(1)}(m)$ in a more compact form. To simplify notation, we assume here that this is not so. ◊

The cost function $\Delta_2^{(1)}(m)$ depends on the unknown parameters $\mathbf{p}_1$. A possible approach is then to maximize det $\mathbf{F}_2^{(1)}(m)$. Consider a generic structure

$$y_m(\xi, \mathbf{p}) = y_{m1}(\xi, \mathbf{p}_1) + y_{m2}(\xi, \mathbf{p}_2),$$

containing all linearly independent terms of the two structures. $\mathbf{F}_2^{(1)}(m)$ then corresponds to $\mathbf{F}_1^g(m)$, the inverse of the covariance matrix of the estimate of $\mathbf{p}_1$ for the generic structure, and det $\mathbf{F}_2^{(1)}(m)$ can be interpreted as a measure of the inadequacy of the second structure for data generated by $y_m(\xi, \mathbf{p})$. An equivalence theorem can be proved, of the same type as for D-optimality (Section 6.2.2.3): provided $\mathbf{F}_{22}(m_{D_s})$ is invertible, the design measure m_{D_s} is D$_s$-optimal if and only if, for all ξ,

$$\mathbf{r}^T(\xi)\mathbf{F}^{-1}(m_{D_s})\mathbf{r}(\xi) - \mathbf{r}_2^T(\xi)\mathbf{F}_{22}^{-1}(m_{D_s})\mathbf{r}_2(\xi) \le n_2,$$

with $\mathbf{r}(\xi)$ the regressor vector of the generic structure. When $\mathbf{F}_{22}(m_{D_s})$ is singular, the cost function $\det \mathbf{F}^{(1)}_{2}(m)$ is not differentiable (and may be discontinuous) at m_{D_s}. An extension of this theorem for that situation can be found in (Silvey, 1980; Pázman, 1986).

Since there is no reason to favour either structure, Atkinson and Cox (1974) suggest the cost function

$$j_{1,2}(m) = [\det \mathbf{F}_2^{g}(m)]^{c_2}[\det \mathbf{F}_1^{g}(m)]^{c_1}, \text{ with } c_1 \text{ and } c_2 > 0,$$

(and, for instance, $c_i = 1/n_i$ to take the dimensions of the two matrices $\mathbf{F}_2^{g}(m)$ and $\mathbf{F}_1^{g}(m)$ into account). This approach easily extends to the case where there are more than two rival structures, by defining a generic structure for the whole set of rival structures, which yields the cost function (Atkinson and Cox, 1974)

$$j_m(m) = \prod_{k=1}^{m} [\det \mathbf{F}_k^{g}(m)]^{1/n_k}.$$

An algorithm of the same type as those used for D- or D_s-optimal design is proposed. Refer to Atwood (1980) and Pázman (1986) for convergence studies in the special case where the optimal experiment is degenerate (*i.e.* does not permit estimation of all parameters of the generic structure).

6.6.3.4 Possible extensions

Joint estimation and discrimination. The most intuitive approach is to address the problems successively, first designing an optimal experiment to choose between structures and then, once a structure has been selected, designing an optimal experiment to estimate its parameters. One may wish to link these two phases more smoothly, to avoid premature selection of an inadequate structure and spend timely effort on estimating the parameters of a suitable structure.

Hill, Hunter and Wichern (1968) suggest a sequential approach based on a linear combination of cost functions for discrimination and estimation. The weight of the former falls monotonically as the largest of the probabilities of the structures increases.

Borth (1975) suggests a cost function based on the notion of total entropy, which includes the structural entropy, already used in the approach presented in Section 6.6.3.2, and the parametric entropy related to the uncertainty in the parameters of the different structures. This idea has been taken up by Huang (1991), who uses the upper bound of Box and Hill (1967) for the expected variation of the structural entropy. A numerical comparison on various simulated examples seems to indicate that the method of Hill, Hunter and Wichern estimates better but discriminates worse than that of Huang.

Other cost functions based on D_s-optimality. A first modification (Huang, 1991) is to replace the cost function of Atkinson and Cox (1974)

$$j_m(m) = \prod_{k=1}^{m} [\det \mathbf{F}_k^{g}(m)]^{1/n_k}$$

by the more intuitively appealing function

$$j(m) = \min_{k \in \{1, \ldots, m\}} [\det \mathbf{F}_k^{\mathrm{g}}(m)]^{1/nk},$$

which yields an approach closely related to T-optimal design.

A second modification (Huang, 1991) consists of replacing the simultaneous comparison of m structures with a unique generic structure by pairwise comparison using simpler generic structures. This may have special interest when m is large, since the unique generic structure may then be rather complicated if the rival structures have few common parts. One thus obtains either a cost function written as a product of determinants (following the approach of Atkinson and Cox), or a maximin cost function (following the modification suggested above).

Numerical examples seem to indicate that the maximin approach requires more computation (which is not surprising), but performs better. Pairwise comparison of structures seems to yield experiments with fewer support points than using a unique generic structure.

Other algorithms. New algorithms for approximate design have been suggested in the context of structure discrimination (Huang, 1991; Huang, Pronzato and Walter, 1991): an exchange algorithm, initially proposed in the literature for the estimation of mixtures (Böhning, 1985, 1989); a global-substitution algorithm (modifying all support points of the design measure at each iteration); an algorithm for the optimization of maximin cost functions (because the direction of steepest ascent no longer corresponds to the introduction of a unique support point into the design measure, contrary to what is indicated in (Atkinson and Fedorov, 1975b)).

An average-optimal approach for T-optimal design. Consider two rival structures. T-optimal design relies on the unrealistic assumption that the true structure (with index i) and the value $\mathbf{p}_i$ of its parameters are known. The cost function is then $\Delta_k^{(i)}(m, \mathbf{p}_i)$, which depends on $\mathbf{p}_i$. The average-optimal approach (Ponce de Leon and Atkinson, 1991a, 1991b), relies on knowledge of the prior probability π_i that the ith structure is true, and the prior density $\pi_{\mathrm{p}}(\mathbf{p}_i)$ of the parameters of this structure (conditional on it being true). The resulting cost function takes the form

$$j(m) = \pi_1 \mathop{\mathrm{E}}_{\mathbf{p}_1} \{\Delta_2^{(1)}(m, \mathbf{p}_1)\} + \pi_2 \mathop{\mathrm{E}}_{\mathbf{p}_2} \{\Delta_1^{(2)}(m, \mathbf{p}_2)\}.$$

This non-sequential approach easily extends to more than two rival structures. An equivalence theorem can again be proved, and used to derive an optimization algorithm similar to those already presented.

6.7 Conclusions

Although experiment design is generally considered an important step of modelling, it is too often restricted to a qualitative study. This is mainly due to some skepticism about quantitative results often based on rather heuristic assumptions. However, although making assumptions is inescapable (any model structure already incorporates some), unrealistically precise prior information can be avoided. Some results deserve emphasis. First, whereas the Fisher information matrix depends on the distribution of the measurement noise, we have seen (in Section 6.1) that the optimal experiment does not, provided that the noise corresponds to a sequence of i.i.d. variables. (Correlated variables have been considered in Section 6.3.2.2.) Second, whereas local design assumes a known prior nominal value for the parameters (when the structure is not LP), the methods presented in Sections 6.4.3 and 6.4.4 allow the uncertainty in this nominal value to be taken into account, and the dependence of the optimal experiment on any other quantity (*e.g.*, a nuisance parameter) with an unreliable prior value could be treated in the same way.

Finally, one should note the simplicity of the optimal experiments usually obtained: repetition of observations under the same experimental conditions, simple input sequences. Their implementation will thus often be easier than that of more conventional and less informative experiments.

7 Falsification

This step is of paramount importance, for it may cause some previous choices to be rejected. It is often called *validation*, which should not be misinterpreted as implying definitive confirmation of the model. In fact, the best one can do is to test the model by trying to falsify (invalidate) it, looking for defects. The aim of this chapter is to describe *some* techniques for such testing. Even if the model successfully passes the tests, its validity remains in doubt, since future tests may lead to its rejection. Most of the techniques are based on analysis of residuals; see, *e.g.* (Anscombe and Tukey, 1963; Draper and Smith, 1981, Cook and Weisberg, 1982). Testing residuals, *e.g.* for homogeneity, stationarity, independence or normality, is among the main topics of applied statistics, so this chapter only gives some guide-lines, and is by no means exhaustive on such a broad subject.

7.1 Simple inspection

First, when the parameters have a physical meaning, they must generally satisfy some inequality constraints (*e.g.* on their sign or order of magnitude). When $\hat{\mathbf{p}}$ is obtained by unconstrained optimization, one can check *a posteriori* whether $\hat{\mathbf{p}}$ is admissible, which may lead to the rejection of the associated model. In behavioural models, when a parameter uncertainty interval (Chapter 5) contains zero, a simplified model structure obtained by removing the associated parameter and the corresponding part of the regressor may be considered.

Testing the predictive capability of the model is often very useful. It consists of comparing the system and model behaviour on a new data set, corrupted by random errors independent of those present when the model was constructed. Inputs (more generally experimental conditions) different from those used for the estimation of the model parameters can also be used (Ljung and Hjalmarsson, 1995). The model will pass the test if it delivers a suitable prediction of the system behaviour associated with these fresh data. This method is especially efficient for discriminating between simple and complex structures, the most complex ones often being unable to reproduce the system behaviour for another sequence of errors (for they model a particular realization of these errors in more detail; see Section 2.5). In structure selection, this procedure is the basis of cross-validation; see *e.g.* (Stone, 1974; Snee, 1977). It requires that not all the data be used to fit the models to the system behaviour. Optimal strategies for partitioning the data records into estimation and validation subsets are discussed in (Djuric and Kay, 1994).

Testing the model for robustness should also be considered. When neglected phenomena may perturb initial conditions, inputs or state variables, or constants taken as known, it is important to check that some small perturbation of these quantities does not drastically modify the behaviour of $\hat{M}(\hat{\mathbf{p}})$. If that were the case, the model would not be robust, and should be used with extreme caution.

Finally, simple graphical analysis of the residuals is also very instructive. Assume, for instance, that

$$y(t_i) = y_{\mathrm{m}}(t_i, \mathbf{p}^*) + \varepsilon(t_i), \quad i = 1, \ldots, n_{\mathrm{t}},$$

where the $\varepsilon(t_i)$'s are independently distributed $\mathcal{N}(0, \sigma_{t_i}^2)$, with $\sigma_{t_i}^2$ known (or parametrized). The evolution of the normalized residuals

$$r_{\mathrm{n}}(t_i) = \frac{y(t_i) - y_{\mathrm{m}}(t_i, \hat{\mathbf{p}})}{\sigma_{t_i}}, \quad i = 1, \ldots, n_{\mathrm{t}},$$

can then be plotted against time. If the number n_{t} of observations is large enough, and if the estimator is consistent, these normalized residuals should resemble the sequence $\varepsilon(t_i)/\sigma_{t_i}$ and thus approximately correspond to i.i.d. $\mathcal{N}(0, 1)$ random variables. This plot may reveal the presence of outliers, a non-zero mean, correlations, non-stationarity (or more generally non-homogeneity, *e.g.*, a sudden and unaccounted-for modification of the experimental conditions). This may in turn lead to rejecting some of the assumptions used in the identification. Plotting the normalized residuals against the inputs may be instructive too, for it may reveal dependence not taken into account by the model structure. Also, plotting histograms of normalized residuals gives a first indication of the validity of the Gaussian assumption; see Section 7.2.1. More sophisticated graphical tools from multivariate data analysis can also be used, see *e.g.* (Atkinson, 1985), and a wealth of software is available.

Such inspection may precede more sophisticated statistical analysis.

7.2 Statistical analysis of residuals

Beyond simple graphical analysis, statistical tools allow the validity of some underlying assumptions to be tested. Outlier detection will not be detailed here. A technique to bypass the problem is to use robust estimation (Sections 3.7.2, 3.7.4 and 5.4.2.2), which yield parameter estimates relatively insensitive to outliers. Their presence *may* then be revealed by inspection of the residuals, making it possible to discard them. More efficient (but less robust) estimation techniques can be used with the data cleaned up in this way.

We shall only consider the analysis of univariate data corresponding to regression residuals (or prediction errors) $e_{\mathrm{p}}(t, \hat{\mathbf{p}})$, with $\hat{\mathbf{p}}$ the estimated value of the parameters (obtained, for instance, by maximum-likelihood estimation). We shall test these residuals against assumptions made about the sequence of perturbations ε(t). Many test statistics can be used, see, *e.g.*, (Kanji, 1993), with specific tabulated distributions. Only procedures that use classical statistical tables (for normal, Student's and Fisher-Snedecor distributions) will be described, testing for normality, stationarity and independence. The validity of these three assumptions is a prerequisite for many other statistical procedures. More details can be found, *e.g.*, in (Madansky, 1988). When the values required for computing test statistics are not available in tables, they may be computed numerically (Press *et al.*, 1986).

EXAMPLE 7.1

The maximum-likelihood approach often relies on the assumption that the prediction error satisfies

$$e_{\mathrm{p}}(t, \mathbf{p}^*) = \varepsilon(t), \quad t = 1, \dots, n_{\mathrm{t}},$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, \sigma^2)$ random variables, with σ^2 not necessarily known. Under the assumption of ergodicity, the mean m_{e} and variance v_{e} of the errors $e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ml}})$ can be estimated by

$$\hat{m}_{\mathrm{e}} = \frac{1}{n_{\mathrm{t}}} \sum_{t=1}^{n_{\mathrm{t}}} e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ml}})$$

and

$$\hat{v}_{\mathrm{e}} = \frac{1}{n_{\mathrm{t}} - 1} \sum_{t=1}^{n_{\mathrm{t}}} [e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ml}}) - \hat{m}_{\mathrm{e}}]^2.$$

If $e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ml}})$ is distributed $\mathcal{N}(m_{\mathrm{e}}, v_{\mathrm{e}})$, $\hat{m}_{\mathrm{e}}$ is distributed $\mathcal{N}(m_{\mathrm{e}}, v_{\mathrm{e}}/n_{\mathrm{t}})$. A classical test for zero mean is then the *t-test*, the test statistic being $t = \hat{m}_{\mathrm{e}}/(\hat{v}_{\mathrm{e}} n_{\mathrm{t}})^{1/2}$. The null hypothesis:

$$\mathrm{H}_0\colon \hat{m}_{\mathrm{e}} = 0,$$

is tested against the alternative hypothesis

$$\mathrm{H}_1\colon \hat{m}_{\mathrm{e}} \neq 0.$$

At significance level α (usually taken as 5%), H_0 will be rejected if $|t| > t_\alpha(n_{\mathrm{t}} - 1)$, where $t_\alpha(n_{\mathrm{t}} - 1)$ has probability $\alpha/2$ of being exceeded by a random variable with Student's t-distribution with $n_{\mathrm{t}} - 1$ degrees of freedom. Critical values of t_α are tabulated in most statistical books. This means that the probability of rejecting H_0 when it is in fact true is α. When n_{t} is large, a simpler test may be used, rejecting H_0 when

$$|\hat{m}_{\mathrm{e}}| > 2\sqrt{\frac{\hat{v}_{\mathrm{e}}}{n_{\mathrm{t}}}}.$$

If H_0 is rejected, the model may be augmented by introducing an additional parameter giving a constant term in the response.

Note that whereas the Gaussian assumption for the distribution of errors is not important here, the assumption of independence is crucial. Indeed, if the central-limit theorem applies and n_{t} is large, then $\hat{m}_{\mathrm{e}}$ is approximately distributed $\mathcal{N}(m_{\mathrm{e}}, v_{\mathrm{e}}/n_{\mathrm{t}})$ even if the $e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ml}})$'s are not Gaussian, provided they are i.i.d. (The assumption that the $e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ml}})$'s are identically distributed is not essential (Rényi, 1966).) ◊

7.2.1 Testing for normality

Many procedures are available. A first consists of comparing the plot of the empirical cumulative distribution function (c.d.f.) F_{e} with that of a normal distribution.

The empirical mean $\hat{m}_{\mathrm{e}}$ and variance $\hat{v}_{\mathrm{e}}$ of the $e_{\mathrm{p}}(t, \hat{\mathbf{p}})$'s are

$$\hat{m}_{\mathrm{e}} = \frac{1}{n_{\mathrm{t}}} \sum_{t=1}^{n_{\mathrm{t}}} e_{\mathrm{p}}(t, \hat{\mathbf{p}})$$

and

$$\hat{v}_{\mathrm{e}} = \frac{1}{n_{\mathrm{t}} - 1} \sum_{t=1}^{n_{\mathrm{t}}} [e_{\mathrm{p}}(t, \hat{\mathbf{p}}) - \hat{m}_{\mathrm{e}}]^2.$$

The errors $e_{\mathrm{p}}(t, \hat{\mathbf{p}})$ are then normalized according to

$$e_{\mathrm{p_n}}(\mathrm{t}, \hat{\mathbf{p}}) = \frac{e_{\mathrm{p}}(t, \hat{\mathbf{p}}) - \hat{m}_{\mathrm{e}}}{\sqrt{\hat{v}_{\mathrm{e}}}}, \quad t = 1, \ldots, n_{\mathrm{t}},$$

and ordered by indexing time instants so that

$$e_{\mathrm{p_n}}(t_1, \hat{\mathbf{p}}) \le e_{\mathrm{p_n}}(t_2, \hat{\mathbf{p}}) \le \ldots \le e_{\mathrm{p_n}}(t_{n_{\mathrm{t}}}, \hat{\mathbf{p}}).$$

The empirical c.d.f. $F_{\mathrm{e}}(x)$ is then

$$F_{\mathrm{e}}(x) = \begin{cases} 0 & \text{if } x < e_{\mathrm{p_n}}(t_1, \hat{\mathbf{p}}), \\ \dfrac{i}{n_{\mathrm{t}}} & \text{if } e_{\mathrm{p_n}}(t_i, \hat{\mathbf{p}}) \le x < e_{\mathrm{p_n}}(t_{i+1}, \hat{\mathbf{p}}), \\ 1 & \text{if } e_{\mathrm{p_n}}(t_{n_{\mathrm{t}}}, \hat{\mathbf{p}}) \le x, \end{cases}$$

which may be plotted against the c.d.f. $F(x)$ of a normal variable $\mathcal{N}(0, 1)$. $F_{\mathrm{e}}(x)$ can also be plotted on *normal paper*, where, under the normality assumption, it should be close to a straight line. The use of such nomograms could be avoided by plotting $F[e_{\mathrm{p_n}}(t_i, \hat{\mathbf{p}})]$ as a function of i/n_{t} (*pp-plot*). Note that

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-\frac{u^2}{2})\, \mathrm{d}u$$

can easily be obtained by numerical integration (Press *et al.*, 1986).

EXAMPLE 7.2

Consider the four following data sets:

— Data Set (i) consists of a sequence of 100 i.i.d. $\mathcal{N}(0, 1)$ random variables;
— Data Set (ii) consists of a sequence of 100 i.i.d. $\mathcal{U}(-1, 1)$ random variables;
— Data Set (iii) consists of a sequence of 100 prediction errors $e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ls}})$, corresponding to residuals of linear regression,

$$e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ls}}) = y(t) - \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}_{\mathrm{ls}},$$

with

$$\hat{\mathbf{p}}_{\mathrm{ls}} = (\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{y}^{\mathrm{s}},$$

where

$$\mathbf{y}^{\mathrm{s}} = \mathbf{R}\mathbf{p}^* + \boldsymbol{\varepsilon},$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix},$$

$$\mathbf{R}_1 = \mathrm{diag}(1, 2, \ldots, 50),$$

$$\mathbf{R}_2 = \mathrm{diag}(51, 52, \ldots, 100),$$

$$\mathbf{p}_i^* = 1, i = 1, \ldots, 50,$$

and $\boldsymbol{\varepsilon}$ is distributed $\mathcal{N}(\mathbf{0}, \mathbf{I}_{100})$;
— Data Set (iv) consists of a sequence of autocorrelated variables $x(t)$, such that

$$x(1) = \varepsilon(1); \quad x(t+1) = -x(t) + \varepsilon(t+1), \quad t = 1, \ldots, 99,$$

where the $\varepsilon(t)$'s are i.i.d. $\mathcal{N}(0, 1)$ random variables.

Figure 7.1 presents a histogram for Data Set (i). The comparison between the cumulative distribution functions $F(x)$ (theoretical) and $F_{\mathrm{e}}(x)$ (empirical) is given in Figure 7.2. The corresponding pp-plot is in Figure 7.3. The histogram does not help much in deciding whether to accept or reject the normality assumption. The decision is easier from Figures 7.2 and 7.3. The pp-plot for Data Set (ii) is given in Figure 7.4. The decision to reject the normality assumption is now easier from the histogram presented in Figure 7.5. The case of Data Set (iii) is more difficult. The residuals $e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ls}})$ are the entries of

$$\mathbf{e}(\hat{\mathbf{p}}_{\mathrm{ls}}) = [\mathbf{I}_{n_{\mathrm{t}}} - \mathbf{R}(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}]\mathbf{y}^{\mathrm{s}},$$

and their covariance matrix is $\mathbf{V} = [\mathbf{I}_{n_{\mathrm{t}}} - \mathbf{R}(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}]$. They are thus generally autocorrelated and non-stationary (heteroscedastic). Since rank $\mathbf{V} = n_{\mathrm{t}} - n_{\mathrm{p}}$, $\mathbf{e}(\hat{\mathbf{p}}_{\mathrm{ls}})$ moves in an $(n_{\mathrm{t}} - n_{\mathrm{p}})$-dimensional space when $\mathbf{y}^{\mathrm{s}}$ varies. When $\boldsymbol{\varepsilon}$ is distributed $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_{n_{\mathrm{t}}})$, as here, $\|\mathbf{e}(\hat{\mathbf{p}}_{\mathrm{ls}})\|^2/\sigma^2$ has a χ^2 distribution with $n_{\mathrm{t}} - n_{\mathrm{p}}$ degrees of freedom. Figures 7.6, 7.7 and 7.8 respectively present the histogram, the functions $F(x)$ and $F_e(x)$, and the pp-plot for this data set.

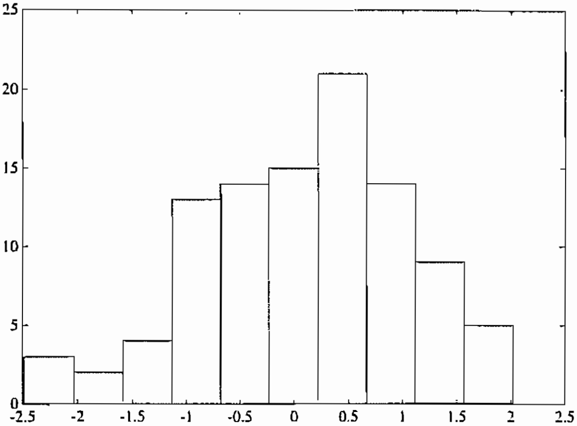


Figure 7.1. Histogram for Data Set (i)

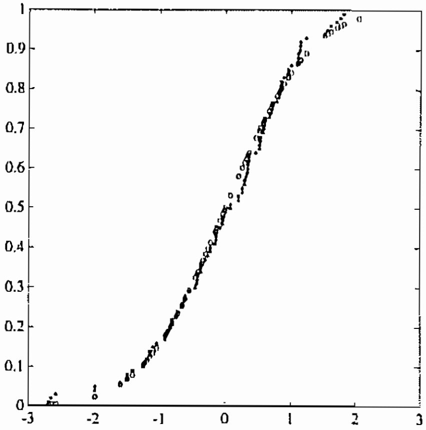


Figure 7.2. Comparison between $F(x)$ (o) and $F_e(x)$ (*) for Data Set (i)

The lack of independence is so marked here as to make the normality assumption seem not to be valid! Caution is thus needed when using such simple graphical techniques if n_t is not very much larger than n_p (here, $n_p = 50 = n_t/2$).

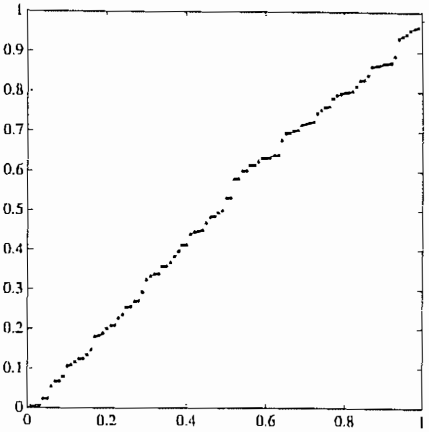


Figure 7.3. pp-plot for Data Set (i)

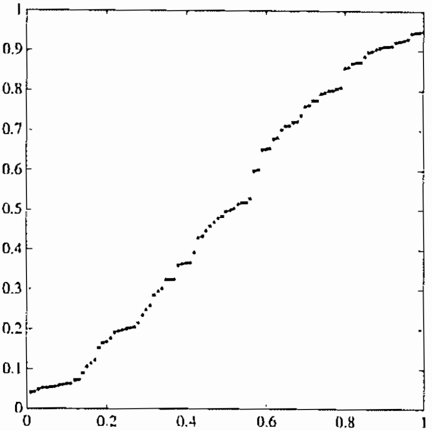


Figure 7.4. pp-plot for Data Set (ii)

When the model structure is LP and least-squares estimation is used, one can construct a sequence of $n_t - n_p$ homoscedastic variables, with covariance matrix $\sigma^2\mathbf{I}_{n_t-n_p}$, by a linear transformation of the observations (provided the $\varepsilon(t)$'s are i.i.d. with zero mean and variance σ^2) (Madansky, 1988, p. 69). This makes it possible to test the $\varepsilon(t)$'s for normality and homoscedasticity even when n_p is not negligible compared with n_t. This procedure is now applied to Data set (iii). First **R** and $\mathbf{y}^s$ are partitioned into

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \end{bmatrix} \quad \text{and} \quad \mathbf{y}^{\mathrm{s}} = \begin{bmatrix} \mathbf{y}_0^{\mathrm{s}} \\ \mathbf{y}_1^{\mathrm{s}} \end{bmatrix},$$

where $\mathbf{R}_0$ is $p \times p$ and nonsingular. The matrix $\mathbf{M} = \mathbf{R}_0(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}_0^{\mathrm{T}}$ is then computed. Denote the eigenvalues of $\mathbf{M}$ smaller than one by $\lambda_1, \ldots, \lambda_n$ and the associated eigenvectors by $\mathbf{v}_1, \ldots, \mathbf{v}_n$. Let $\mathbf{e}_0(\hat{\mathbf{p}}_{\mathrm{ls}})$ and $\mathbf{e}_1(\hat{\mathbf{p}}_{\mathrm{ls}})$ be the residuals associated with the partition of $\mathbf{R}$ and $\mathbf{y}^{\mathrm{s}}$, that is

$$\mathbf{e}_0(\hat{\mathbf{p}}_{\mathrm{ls}}) = \mathbf{y}_0^{\mathrm{s}} - \mathbf{R}_0\hat{\mathbf{p}}_{\mathrm{ls}},$$

and

$$\mathbf{e}_1(\hat{\mathbf{p}}_{\mathrm{ls}}) = \mathbf{y}_1^{\mathrm{s}} - \mathbf{R}_1\hat{\mathbf{p}}_{\mathrm{ls}}.$$

The $n_{\mathrm{t}} - n_{\mathrm{p}}$ variables to be considered are given by

$$\mathbf{e}' = \mathbf{e}_1(\hat{\mathbf{p}}_{\mathrm{ls}}) - \mathbf{R}_1\mathbf{R}_0^{-1}\Big(\sum_{i=1}^{n} \frac{\sqrt{\lambda_i}}{1 + \sqrt{\lambda_i}}\, \mathbf{v}_i\mathbf{v}_i^{\mathrm{T}}\Big)\, \mathbf{e}_0(\hat{\mathbf{p}}_{\mathrm{ls}}).$$

They form Data Set (iii'), which will be tested for normality, stationarity and independence (with $n_{\mathrm{p}} = 0$).

Figure 7.9 presents the distribution functions $F_{\mathrm{e}}(x)$ and $F(x)$ for Data Set (iii') and Figure 7.10 the corresponding pp-plot. Comparison with Figures 7.7 and 7.8 shows that normality has been much improved by the transformation.

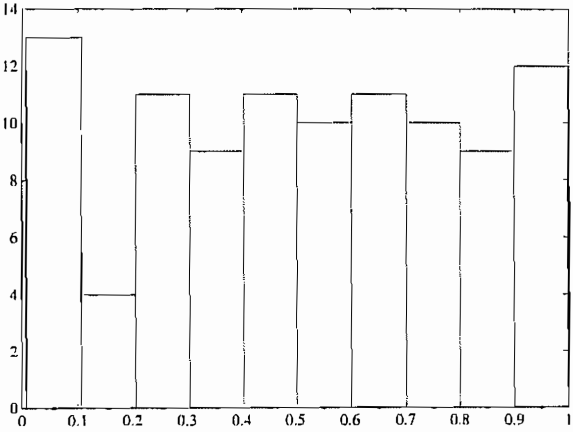


Figure 7.5. Histogram for Data Set (ii)

For autocorrelated Data Set (iv), $F(x)$ and $F_{\mathrm{e}}(x)$ are presented in Figure 7.11 and the pp-plot is in Figure 7.12. Note the similarity between Figures 7.7 and 7.11 (or 7.8 and 7.12). ◊

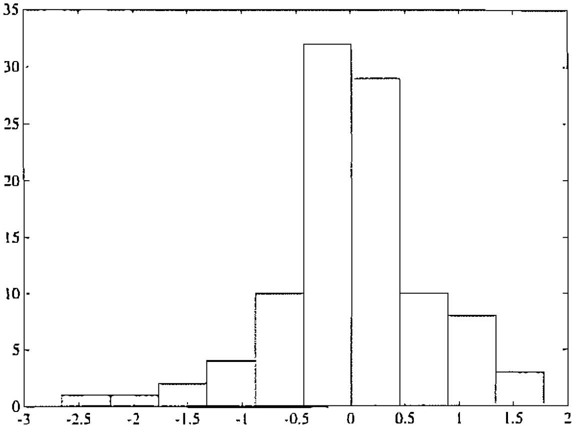


Figure 7.6. Histogram for Data Set (iii)

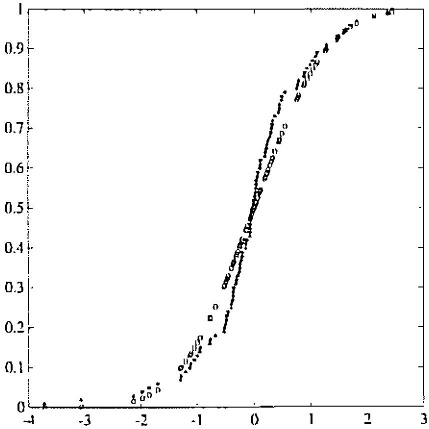


Figure 7.7. Comparison between $F(x)$ (o) and $F_e(x)$ (*) for Data Set (iii)

The graphical procedure above can be completed by statistical tests. For instance, the *Kolmogorov-Smirnov test* uses the maximum difference between $F_e(x)$ and $F(x)$ as a test statistic. The values exceeded with probability α by

$$\max_i |F_e[e_{p_n}(t_i, \hat{\mathbf{p}})] - F[e_{p_n}(t_i, \hat{\mathbf{p}})]| = \max_i |\frac{i}{n_t} - F[e_{p_n}(t_i, \hat{\mathbf{p}})]|$$

under the normality assumption are tabulated for various values of n_t. Similarly, the *Shapiro-Wilk, Filiben and D'Agostino tests* rely on the regression of $F^{-1}(i/n_t)$ as a function of $e_{p_n}(t_i, \hat{\mathbf{p}})$. Again, comparisons of test statistics with tabulated values determine acceptance or rejection of the normality assumption.

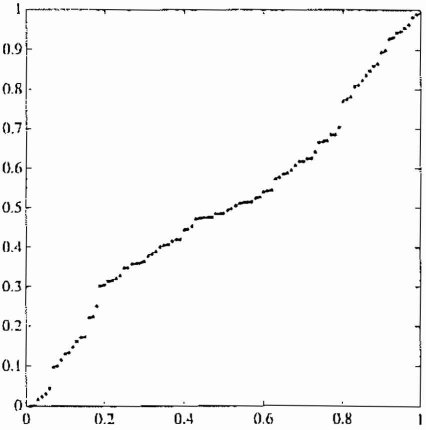


Figure 7.8. pp-plot for Data Set (iii)

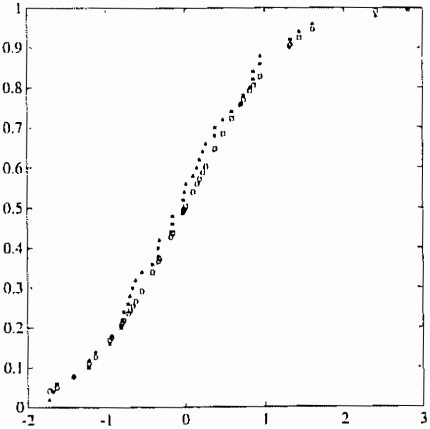


Figure 7.9. Comparison between $F(x)$ and $F_e(x)$ for Data Set (iii')

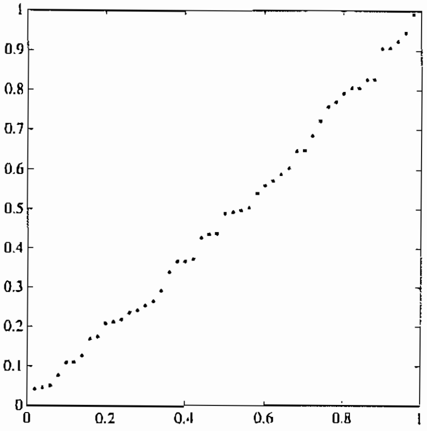


Figure 7.10. pp-plot for Data Set (iii')

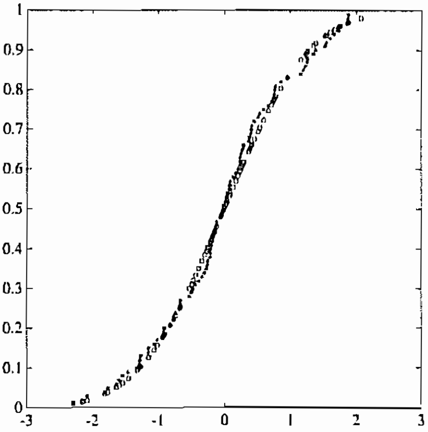


Figure 7.11. Comparison between $F(x)$ (o) and $F_e(x)$ (*) for Data Set (iv)

Empirical moments can also be used to test data for normality. Let $X(t)$ ($t = 1, \ldots , n_t$) be an i.i.d. sequence of random variables, with probability density function $f(x)$. The test statistic for skewness

$$\gamma_1 = \frac{\mathrm{E}\{(X - \mathrm{E}\{X\})^3\}}{\sigma^3}$$

is based on the third-order moment and that for kurtosis

$$\gamma_2 = \frac{\mathrm{E}\{(X - \mathrm{E}\{X\})^4\}}{\sigma^4} - 3$$

on the fourth-order moment.

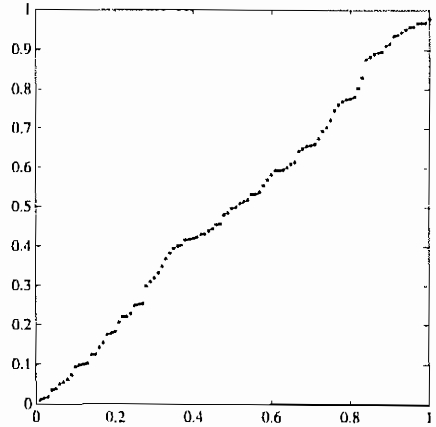


Figure 7.12. pp-plot for Data Set (iv)

When the distribution is symmetric about the mean $\mathrm{E}\{X\}$, $\gamma_1 = 0$, and

$$\gamma_1 > 0 \text{ if } \int_{\mathrm{E}\{X\}}^{\infty} f(x)\mathrm{d}x > \int_{-\infty}^{\mathrm{E}\{X\}} f(x)\mathrm{d}x,$$

$$\gamma_1 < 0 \text{ if } \int_{\mathrm{E}\{X\}}^{\infty} f(x)\mathrm{d}x < \int_{-\infty}^{\mathrm{E}\{X\}} f(x)\mathrm{d}x.$$

For a normal distribution, $\gamma_1 = \gamma_2 = 0$. If $f(x)$ decreases more rapidly than the density of the normal distribution as x tends to $\pm\infty$, $\gamma_2 < 0$. If $f(x)$ decreases more slowly, $\gamma_2 > 0$. The test statistics are then the empirical values

$$\hat{\gamma}_1 = \frac{\frac{1}{n_t}\sum_{t=1}^{n_t} [e_{\mathrm{p}}(t, \hat{\mathbf{p}}) - \hat{m}_{\mathrm{c}}]^3}{\hat{v}_{\mathrm{c}}^{3/2}},$$

and

$$\hat{\gamma}_2 = \frac{\dfrac{1}{n_t}\displaystyle\sum_{t=1}^{n_t}[e_p(t,\hat{\mathbf{p}}) - \hat{m}_e]^4}{\hat{v}_e^2} - 3.$$

Under the normality assumption, the means of $\hat{\gamma}_1$ and $\hat{\gamma}_2$ are respectively

$$m_1 = 0 \quad \text{and} \quad m_2 = -\frac{6}{n_t + 1},$$

and their variances

$$v_1 = \frac{6(n_t - 2)}{(n_t + 1)(n_t + 3)} \quad \text{and} \quad v_2 = \frac{24n_t(n_t - 2)(n_t - 3)}{(n_t + 1)^2(n_t + 3)(n_t + 5)}.$$

Moreover, the asymptotic distributions (as $n_t \to \infty$) of $\hat{\gamma}_1$ and $\hat{\gamma}_2$ are normal. The normality assumption will thus be rejected (at significance level 5%) if

$$T_1 = \frac{|\hat{\gamma}_1|}{2\sqrt{v_1}} > 1$$

or

$$T_2 = \frac{|\hat{\gamma}_2 - m_2|}{2\sqrt{v_2}} > 1,$$

since $\text{prob}(T_1 < 1) = \text{prob}(T_2 < 1) = 95\%$ for normal variates.

Another method relies on the comparison between the sample range w of the errors $e_p(t, \hat{\mathbf{p}})$ and standard deviation $\hat{v}_e^{1/2}$. The test statistic is $w/(\hat{v}_e)^{1/2}$, the critical values of which are tabulated (Kanji, 1993).

EXAMPLE 7.2 (continued)

The values of the test statistics T_1 and T_2 for the five data sets are indicated in Table 7.1.

Data Set	T_1	T_2
(i)	0.944	0.083
(ii)	0.172	1.197
(iii)	1.019	2.286
(iii')	0.815	0.296
(iv)	0.057	0.389

Table 7.1. Test statistics for normality

Data Set (i) passes tests T_1 and T_2. For the uniformly distributed Data Set (ii), T_1 confirms that the distribution is symmetric, but the normality assumption is rejected by T_2. The residuals of Data Set (iii) are too correlated to obtain a correct decision from these tests. The condition $n_t >> n_p$ is again seen to be essential for normality testing by this approach. The corrected residuals of Data Set (iii') pass the tests. The decisions concerning Data Set (iv) are correct. ◊

7.2.2 Testing for stationarity

Although stationarity is generally taken as with respect to time, it can also cover homogeneity with respect to other independent variables. We shall only consider second-order stationarity, that is constancy of the variance σ^2 (or homoscedasticity). Many procedures for statistical analysis rely on this assumption. Moreover, knowing that it should be rejected may suggest a change of the estimation criterion, for instance from unweighted to weighted least-squares. Again, we shall only present methods that do not require the use of very specific tables.

The *Goldfeld-Quandt procedure* relies on the intuitive idea of splitting the data set into three parts. The first k_1 and last k_3 data points are kept, while the k_2 intermediate data points are eliminated. A rule of thumb is $k_1 = k_3$ and $k_2 = n_t/4$. The first k_1 and last k_3 errors $e_p(t, \hat{\mathbf{p}})$ are then used to compute the ratio

$$r_3 = \frac{k_3 - n_p}{k_1 - n_p} \frac{\sum_{t=1}^{k_1} [e_p(t, \hat{\mathbf{p}}) - \hat{m}_{e_1}]^2}{\sum_{t=n_t-k_3+1}^{n_t} [e_p(t, \hat{\mathbf{p}}) - \hat{m}_{e_2}]^2},$$

where the means $\hat{m}_{e_1}$ and $\hat{m}_{e_2}$ are computed for the first k_1 and last k_3 points. For an LP structure with unweighted least-squares estimation, and if we assume an i.i.d. normal measurement noise, r_3 has a Fisher-Snedecor distribution with $k_1 - n_p$ and $k_3 - n_p$ degrees of freedom, which we denote by $\mathcal{F}(k_1 - n_p, k_3 - n_p)$. The critical values of r_3 are tabulated in most statistical books. The null hypothesis H_0 is that the variance is the same for the first k_1 and last k_3 data points. If the alternative hypothesis H_1 is that the variance decreases with k,

$$T_3 = \frac{r_3}{F_{0.05}(k_1 - n_p, k_3 - n_p)}$$

is computed, where $F_{0.05}(k_1 - n_p, k_3 - n_p)$ has probability 0.05 of being exceeded by a random variable distributed $\mathcal{F}(k_1 - n_p, k_3 - n_p)$, and H_0 is rejected if $T_3 > 1$. If the alternative hypothesis is that the variance increases with k,

$$T_3' = \frac{r_3}{F_{0.95}(k_1 - n_p, k_3 - n_p)}$$

is computed, and H_0 rejected if $T_3' < 1$. If the alternative hypothesis does not specify which way the variance varies with k, a test (based of the lower and upper tails of the $\mathcal{F}$-distribution) more accurate than successive use of T_3 and T_3' can be used (Madansky, 1988).

Another approach is based on *regression of the squares of residuals* on some explanatory exogenous variables $z_i(t)$ (*e.g.* $z_1(t) = t$, or, when the initial estimation problem corresponds to linear regression with regressors $\mathbf{r}(t)$, $\mathbf{z}(t) = \mathbf{r}(t)$). Consider the normalized residuals

$$e_{\mathrm{p_n}}(t, \hat{\mathbf{p}}) = \frac{e_{\mathrm{p}}(t, \hat{\mathbf{p}}) - \hat{m}_{\mathrm{e}}}{\sqrt{\hat{v}_{\mathrm{e}}}}, \, t = 1, \dots, n_{\mathrm{t}},$$

and define $\nu_{\mathrm{p_n}}(t, \hat{\mathbf{p}}) = e^2_{\mathrm{p_n}}(t, \hat{\mathbf{p}})$. We use the linear regression model

$$\nu_{\mathrm{p_n}}(t, \hat{\mathbf{p}}) = \alpha_0 + \sum_{i=1}^{\dim \mathbf{z}} \alpha_i z_i(t) + \varepsilon'(t).$$

Under the stationarity hypothesis, the α_i's ($i = 1, \dots, \dim \mathbf{z}$) should be zero. The vector

$$\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \dots, \alpha_{\dim \mathbf{z}})^{\mathrm{T}}$$

can be estimated (by least-squares) as

$$\hat{\boldsymbol{\alpha}} = (\mathbf{Z}^{\mathrm{T}}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{v}_{\mathrm{p_n}}(\hat{\mathbf{p}}),$$

where

$$\mathbf{v}_{\mathrm{p_n}}(\hat{\mathbf{p}}) = [\nu_{\mathrm{p_n}}(1, \hat{\mathbf{p}}), \nu_{\mathrm{p_n}}(2, \hat{\mathbf{p}}), \dots, \nu_{\mathrm{p_n}}(n_{\mathrm{t}}, \hat{\mathbf{p}})]^{\mathrm{T}},$$

and where the first column of $\mathbf{Z}$ contains only ones, and the $(i + 1)$th column is $[z_i(1), z_i(2), \dots, z_i(n_{\mathrm{t}})]^{\mathrm{T}}$. It is also instructive to plot $\nu_{\mathrm{p_n}}(t, \hat{\mathbf{p}})$ against the $z_i(t)$'s (see Figure 7.13).

The *Lagrange multiplier test* relies on the computation of the sum of squares of residuals explained by the regression over the $z_i(t)$'s. The sum of squares of residuals corresponding to the regression model above is

$$\| \mathbf{v}_{\mathrm{p_n}}(\hat{\mathbf{p}}) - \mathbf{Z}\hat{\boldsymbol{\alpha}} \|_2^2 = \sum_{t=1}^{n_{\mathrm{t}}} \nu^2_{\mathrm{p_n}}(\mathrm{t}, \hat{\mathbf{p}}) - \mathbf{v}^{\mathrm{T}}_{\mathrm{p_n}}(\hat{\mathbf{p}})\mathbf{Z}(\mathbf{Z}^{\mathrm{T}}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{v}_{\mathrm{p_n}}(\hat{\mathbf{p}}).$$

When only α_0 is estimated, with $\alpha_i = 0$ for $i \neq 0$, the residual sum of squares is

$$\sum_{t=1}^{n_{\mathrm{t}}} \nu^2_{\mathrm{p_n}}(t, \hat{\mathbf{p}}) - \frac{1}{n_{\mathrm{t}}}\left(\sum_{t=1}^{n_{\mathrm{t}}} \nu_{\mathrm{p_n}}(t, \hat{\mathbf{p}})\right)^2.$$

The residual sum of squares explained by the $z_i(t)$'s is therefore

$$s_4 = \mathbf{v}^{\mathrm{T}}_{\mathrm{p_n}}(\hat{\mathbf{p}})\mathbf{Z}(\mathbf{Z}^{\mathrm{T}}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{v}_{\mathrm{p_n}}(\hat{\mathbf{p}}) - \frac{1}{n_{\mathrm{t}}}\left(\sum_{t=1}^{n_{\mathrm{t}}} \nu_{\mathrm{p_n}}(t, \hat{\mathbf{p}})\right)^2.$$

It is to be compared to the total sum of squares

$$s_4' = \sum_{t=1}^{n_t} v_{\mathrm{p_n}}^2(t, \hat{\mathbf{p}}).$$

The ratio

$$r_4 = \frac{s_4}{s_4'},$$

should be much less than one when stationarity with respect to the z_i's is satisfied. Breusch and Pagan (1979) have shown that $s_4/2$ has a χ^2 distribution with dim $\mathbf{z}$ degrees of freedom. A test statistic is thus

$$T_4 = \frac{s_4}{2\chi^2_{0.05}(\dim \mathbf{z})},$$

with $\chi^2_{0.05}(\dim \mathbf{z})$ having probability 5% of being exceeded by a random variable distributed $\chi^2(\dim \mathbf{z})$. The assumption of stationarity with respect to the z_i's will be rejected when $T_4 > 1$.

Similar ideas lead to evaluating the correlation between the variables $v_{\mathrm{p_n}}(t, \hat{\mathbf{p}})$ and $z_i(t)$, that is

$$c_5(i) = \frac{\sum_{t=1}^{n_t} [v_{\mathrm{p_n}}(t, \hat{\mathbf{p}}) - m_{\mathrm{v}}][z_i(t) - m_{\mathrm{z}_i}]}{\left(\sum_{t=1}^{n_t} [v_{\mathrm{p_n}}(t, \hat{\mathbf{p}}) - m_{\mathrm{v}}]^2 \sum_{t=1}^{n_t} [z_i(t) - m_{\mathrm{z}_i}]^2\right)^{1/2}},$$

where m_{v} and m_{z_i} respectively denote the means of $v_{\mathrm{p_n}}(t, \hat{\mathbf{p}})$ and $z_i(t)$. This ratio always satisfies $-1 \leq c_5(i) \leq 1$, and $c_5(i)$ is close to zero when stationarity with respect to z_i is satisfied. This corresponds to *Anscombe's test* (Anscombe, 1961): $c_5(i)(n_t - n_{\mathrm{p}})^{1/2}$ is approximately distributed $\mathcal{N}(0, 1)$, so a test statistic is

$$T_5(i) = \frac{|c_5(i)|\sqrt{n_t - n_{\mathrm{p}}}}{2}.$$

The assumption of stationarity with respect to z_i will then be rejected when $T_5(i) > 1$.

EXAMPLE 7.2 (continued)

We consider now three additional data sets, residuals $e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ls}})$ of linear regression:

$$e_{\mathrm{p}}(t, \hat{\mathbf{p}}_{\mathrm{ls}}) = y(t) - \mathbf{r}^{\mathrm{T}}(t)\hat{\mathbf{p}}_{\mathrm{ls}},$$

with

$$\hat{\mathbf{p}}_{\mathrm{ls}} = (\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{y}^{\mathrm{s}},$$

where

$$\mathbf{y}^s = \mathbf{R}\mathbf{p}^* + \boldsymbol{\varepsilon},$$

$$\mathbf{R} = (\mathbf{v}_1, \mathbf{v}_2),$$

$$v_{1_i} = \cos\left(\frac{i-1}{99}\frac{\pi}{2}\right),\ i = 1, \dots, 100,$$

$$v_{2_i} = \sin\left(\frac{i-1}{99}\frac{\pi}{2}\right),\ i = 1, \dots, 100,$$

$$\mathbf{p}^* = (1, 1)^{\mathrm{T}}.$$

These three data sets differ in the values of $\boldsymbol{\varepsilon}$.

- — In Data Set (v), the $\varepsilon(t)$'s are a sequence of 100 i.i.d. $\mathcal{N}(0, 1)$ random variables.
- — In Data Set (vi), $\boldsymbol{\varepsilon}$ is distributed $\mathcal{N}(\mathbf{0}, \mathbf{C}_1)$, with $\mathbf{C}_1 = \mathrm{diag}(1, 2^2, \dots, 100^2)$, which makes the distribution of the $\varepsilon(t)$'s (very) non-stationary.
- — In Data Set (vii), $\boldsymbol{\varepsilon}$ is distributed $\mathcal{N}(\mathbf{0}, \mathbf{C}_2)$, with $\mathbf{C}_2 = \mathbf{M}\mathbf{C}_1\mathbf{M}^{\mathrm{T}}$, and $\mathrm{M}_{i,i} = \mathrm{M}_{i,i+1} = 1$, $\mathrm{M}_{i,k} = 0\ \forall\ k \neq i, i+1$, which makes the $\varepsilon(t)$'s serially dependent and (very) non-stationary.

Figure 7.13 gives $\nu_{\mathrm{p_n}}(t, \hat{\mathbf{p}}_{\mathrm{ls}})$ and the response of the regression model

$$\nu_{\mathrm{m}}(t, \hat{\boldsymbol{\alpha}}) = \hat{\alpha}_0 + \sum_{i=1}^{2} \hat{\alpha}_i z_i(t)$$

as functions of t, with $z_1(t) = v_{1_t}$ and $z_2(t) = v_{2_t}$ for Data Set (vi).

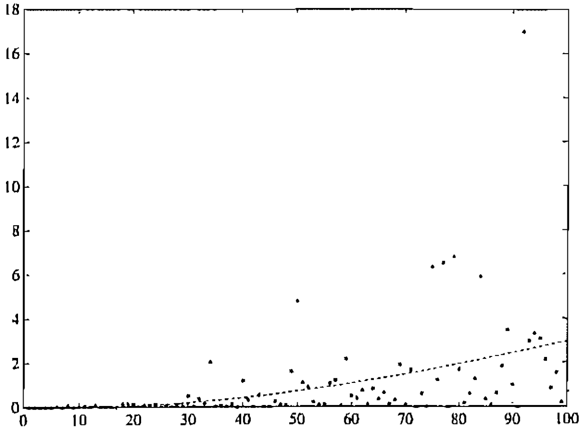


Figure 7.13. $\nu_{\mathrm{p_n}}(t, \hat{\mathbf{p}})$ and model response $\nu_{\mathrm{m}}(t, \hat{\boldsymbol{\alpha}})$ for Data Set (vi)

Non-stationarity is clear from this figure. The values of T_3, T'_3, r_4, T_4, c_5 and T_5 for Data Sets (iii') and (v) to (vii) are given in Table 7.2. (T_3 and T'_3 cannot be used for Data Set (iii), because n_p is too large compared with n_t, whereas n_p is taken equal to zero for Data Set (iii').) For tests T_4 and T_5, the explanatory variables were $z_1(t) = t$ for Data Sets (iii) and (iii'), and $z_1(t) = v_{1_t}$ and $z_2(t) = v_{2_t}$ for Data Sets (v) to (vii).

Data Set	T_3	T'_3	r_4	T_4	$c_5(1)$	$T_5(1)$	$c_5(2)$	$T_5(2)$
(v)	0.502	1.520	0.002	0.056	–0.058	0.287	0.051	0.254
(vi)	0.036	0.110	0.151	7.144	–0.427	2.112	0.381	1.886
(vii)	0.006	0.019	0.229	10.49	–0.523	2.587	0.449	2.223
(iii)	—	—	0.065	4.150	–0.285	1.006	—	—
(iii')	0.303	1.425	0.013	0.264	0.138	0.486	—	—

Table 7.2. Tests statistics for stationarity

The conclusions are correct for the three data sets (v), (vi) and (vii) and the four tests T_3, T'_3, T_4 and T_5. Note that the variables $z_1(t)$ and $z_2(t)$ for tests T_4 and T_5 are oscillating functions, and non-stationarity would be even more easily detected using time as the explanatory variable, as for Data Sets (iii) and (iii'). The residuals of Data Set (iii) does not pass T_4 and T_5, whereas the corrected residuals of Data Set (iii') pass T_3, T'_3, T_4 and T_5. ◊

7.2.3 Testing for independence

The absence of correlation is a necessary condition for independence, and is also sufficient in the case of normal variables. The sample autocorrelation of the prediction errors is given by

$$\hat{c}_\mathrm{e}(k) = \frac{1}{n_\mathrm{t} - k} \sum_{t=1}^{n_\mathrm{t}-k} [e_\mathrm{p}(t, \hat{\mathbf{p}}_\mathrm{ml}) - \hat{m}_\mathrm{e}][e_\mathrm{p}(t+k, \hat{\mathbf{p}}_\mathrm{ml}) - \hat{m}_\mathrm{e}]$$

and their normalized sample autocorrelation is

$$\hat{c}_\mathrm{n}(k) = \frac{\hat{c}_\mathrm{e}(k)}{\hat{c}_\mathrm{e}(0)}.$$

Under the hypothesis of independence (H_0), when n_t tends to infinity, the distribution of $\hat{c}_\mathrm{n}(k)$ should resemble the normal distribution $\mathcal{N}(0, 1/n_\mathrm{t})$ for any $k \neq 0$. One can thus plot $\hat{c}_\mathrm{n}(k)$, which, for n_t large enough, should lie in the interval $[-2/\sqrt{n_\mathrm{t}}, 2/\sqrt{n_\mathrm{t}}]$ with probability close to 95%. However, this does not take into account the fact that the number of data points used to evaluate $\hat{c}_\mathrm{n}(k)$ varies with k. If the variance of $\hat{c}_\mathrm{n}(k)$ is approximated by $n_\mathrm{t}(n_\mathrm{t} + 2)^{-1}(n_\mathrm{t} - k)^{-1}$ instead of $1/n_\mathrm{t}$, $\hat{c}_\mathrm{n}(k)$ should lie in the interval

$$\mathbb{I}(k) = \left[- \frac{2\sqrt{n_\mathrm{t}}}{\sqrt{(n_\mathrm{t} + 2)(n_\mathrm{t} - k)}}, \frac{2\sqrt{n_\mathrm{t}}}{\sqrt{(n_\mathrm{t} + 2)(n_\mathrm{t} - k)}}\right]$$

with probability close to 95%. Note that the size of $\mathbb{I}(k)$ increases with k.

EXAMPLE 7.2 (continued)

Figure 7.14 presents the normalized autocorrelation $\hat{c}_{\mathrm{n}}(k)$ for Data Set (vii), together with the bounds of the interval $\mathbb{I}(k)$ defined above, as functions of k. These bounds are crossed several times, which raises some doubts about the validity of the hypothesis of independence (actually not satisfied). ◊

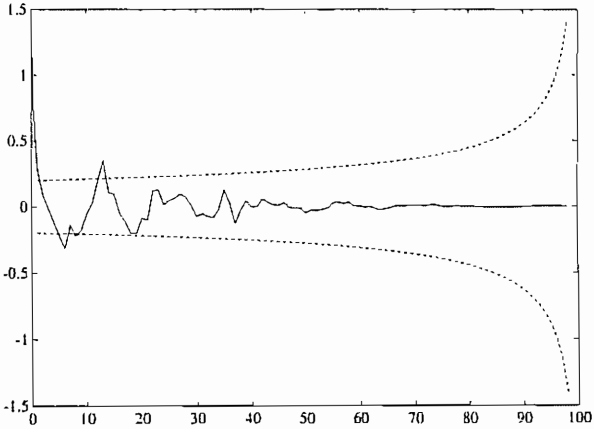


Figure 7.14. Normalized sample autocorrelation $\hat{c}_{\mathrm{n}}(k)$ and bounds of the interval $\mathbb{I}(k)$ as functions of k for Data Set (vii).

Moreover, when n_{t} is large, the variable

$$\nu_6 = \frac{n_{\mathrm{t}} + 2}{n_{\mathrm{t}}} \sum_{k=1}^{n_{\mathrm{k}}} (n_{\mathrm{t}} - k)\hat{c}_{\mathrm{n}}^2(k)$$

approximately has a χ^2 distribution with $n_{\mathrm{k}} - n_{\mathrm{p}}$ degrees of freedom. One can thus use the ratio

$$T_6 = \frac{\nu_6}{\chi_{0.05}^2(n_{\mathrm{k}} - n_{\mathrm{p}})}$$

as a test statistic, where $\chi_{0.05}^2(n_{\mathrm{k}} - n_{\mathrm{p}})$ has probability 0.05 of being exceeded by a random variable distributed $\chi^2(n_{\mathrm{k}} - n_{\mathrm{p}})$. The hypothesis of independence will be rejected when $T_6 > 1$.

Note that correlation between past inputs and residuals can also be used for model validation. The importance of this statistic is stressed in (Ljung and Hjalmarsson, 1995), from an inductive point of view with strong intuitive appeal.

Various non-parametric procedures also permit a sequence of random variables to be tested for independence. They generally rely on the intuitive idea that, when independence holds, each variable in the sequence has probability 1/2 of being larger (or smaller) than the median m, and has probability 1/2 of being larger (or smaller) than the previous variable in the sequence.

With each variable $e_p(t, \hat{\mathbf{p}})$, we associate a new variable $x(t)$:

$$x(t) = \begin{cases} 1 \text{ if } e_p(t, \hat{\mathbf{p}}) > m\,, \\ 0 \text{ if } e_p(t, \hat{\mathbf{p}}) < m\,, \end{cases}$$

(and $x(t) = 0$ or 1 with probability 1/2 if $e_p(t, \hat{\mathbf{p}}) = m$). A first test statistic based on the sequence of $x(t)$'s is the *run test*. A run is a succession of elements with the same value. Let r denote the number of runs in the sequence. For instance, if the sequence of $x(t)$'s is 0110001001110, $r = 7$. Let n_0 denote the number of ones in the sequence $x(t)$ ($n_0 = 6$ in this example). When the observations in the original data set are independent of the order in the sequence, and when n_t is large, the distribution of r for a given value of n_0 is approximately normal, with mean

$$m_7 = \frac{2n_0(n_t - n_0)}{n_t} + 1\,,$$

and variance

$$v_7 = \frac{2n_0(n_t - n_0)[2n_0(n_t - n_0) - n_t]}{n_t^2(n_t - 1)}.$$

The test statistic will thus be

$$T_7 = \frac{|r - m_7|}{2\sqrt{v_7}},$$

and independence will be rejected if $T_7 > 1$.

REMARK 7.1

This non-parametric test is used here *a posteriori* to check the validity of a model. One could also use maximization of the number of runs as a criterion to estimate **p** (Section 3.7.4). ◊

Another procedure considers the successive differences in the sequence of errors, that is

$$x(t) = \begin{cases} 1 \text{ if } e_p(t+1, \hat{\mathbf{p}}) > e_p(t, \hat{\mathbf{p}}), \\ 0 \text{ if } e_p(t+1, \hat{\mathbf{p}}) < e_p(t, \hat{\mathbf{p}}), \end{cases}$$

(with $x(t) = 0$ or 1 with probability 1/2 if $e_p(t+1, \hat{\mathbf{p}}) = e_p(t, \hat{\mathbf{p}})$). When the observations in the original data set are independent of the order in the sequence, and when n_t is large, the number r of runs in the sequence of $x(t)$'s is then approximately normal, with mean

$$m_8 = \frac{2n_t - 1}{3},$$

and variance

$$\nu_8 = \frac{16n_t - 29}{90}.$$

Note in particular that $r = 1$ corresponds to a monotonic sequence of errors $e_p(t, \hat{\mathbf{p}})$, whereas $r = n_t - 1$ corresponds to a sequence which oscillates in direction with period two. The test statistic is then

$$T_8 = \frac{|r - m_8|}{2\sqrt{\nu_8}},$$

and the hypothesis of independence will be rejected when $T_8 > 1$.

A last approach determines the ranking order $r(t)$ of each error $e_p(t, \hat{\mathbf{p}})$ (in increasing or decreasing order), and calculates the *Von-Neumann ratio*:

$$r_9 = \frac{12\sum_{t=2}^{n_t}[r(t) - r(t-1)]^2}{n_t(n_t^2 - 1)},$$

the statistics of which are tabulated. When n_t is large, and under the hypothesis of independence, the distribution of r_9 is approximately normal $\mathcal{N}(2, 20/(5n_t + 7))$. The test statistic will thus be

$$T_9 = \frac{|r_9 - 2|}{2\sqrt{\dfrac{20}{5n_t + 7}}},$$

and the hypothesis of independence will be rejected if $T_9 > 1$.

EXAMPLE 7.2 (continued)

Consider again Data Sets (i-vii). The test statistics T_6 to T_9 are given in Table 7.3.

Data Set	T_6	T_7	T_8	T_9
(i)	0.578	0	0.878	0.655
(ii)	0.426	0.804	1.276	0.562
(iii)	1.056	0.804	2.792	0.560
(iii')	0.584	0.571	2.050	0.517
(iv)	7.896	7.237	5.425	9.369
(v)	0.738	0	0.878	0.655
(vi)	0.923	0	0.878	0.593
(vii)	2.056	2.814	2.952	3.074

Table 7.3. Test statistics for independence

All confirm that Data Sets (iv) and (vii) are not independent. The lack of independence in the residuals of Data Set (iii) is detected by T_6 and T_8 (which explains the poor performance of normality tests T_1 and T_2). The situation is improved for the corrected residuals of Data Set (iii'). The non-stationarity of Data Set (vi) has no effect on the performance of the four tests. ◊

7.3 Conclusions

The statistical tests presented here do not always give an unambiguous conclusion about the validity of the hypothesis tested. If doubt remains, fictitious data can be generated, by simulation of the model response using the estimated values of the parameters, corrupted by perturbations satisfying the tested hypothesis. The same testing procedure applied to these simulated data should then reveal the same ambiguity. If there is clear evidence that the hypothesis is true for the simulated data, it could mean that some important aspect of the behaviour of the actual system has been neglected, and that the hypothesis should be rejected.

Quite often there are several competing model structures for a single process. Proving superiority of a given model is of course an effective way of eliminating the others. Defects which are significant for some applications may be of minor importance for others. Note, however, that defining a quantitative criterion related to the final purpose is not always easy (and a similar problem has been encountered in Chapter 3).

Finally, if the available data do not permit selection of one of the rival models, one should try to collect more data (provided the structures considered are distinguishable). Experimental design for model discrimination should then be considered (as in Section 6.6.3).

References

Adams, E., and U. Kulish (eds.) (1993). *Scientific Computing with Automatic Result Verification*, Mathematics in Sciences and Engineering, **189**, Academic Press, New York.

Albert, A. (1972). *Regression and the Moore-Penrose Pseudoinverse*, Academic Press, New York.

Amari, S. (1985). *Differential-Geometrical Methods in Statistics*, Springer-Verlag, Berlin.

Anderson, B.D.O. (1985). Identification of scalar errors-in-variables models with dynamics, *Automatica*, **21**, 709-716.

Anderson, B.D.O., and J.B. Moore (1979). *Optimal Filtering*, Prentice-Hall, Englewood Cliffs.

Anscombe, F.J. (1961). Examination of residuals, in *Proc. Fourth Berkeley Symposium on Mathematical Statistics and Probability* (J. Neyman, ed.), **1**, University of California Press, Berkeley, 1-36.

Anscombe, F.J. and J.W. Tukey (1963). The examination and analysis of residuals, *Technometrics*, **5**: 141-160.

Arnold, S.F. (1981). *The Theory of Linear Models and Multivariate Analysis*, Wiley, New York.

Arruda, L.V.R., and G. Favier (1991). A review and a comparison of robust estimation methods, in *Prep. 9th IFAC/IFORS Symp. on Identification and System Parameter Estimation* (Cs. Bânyâsz and L. Keviczky, eds.), Budapest, 1027-1032.

Åström, K.J., U. Borisson, L. Ljung and B. Wittenmark (1977). Theory and applications of self-tuning regulators, *Automatica*, **13**, 457-476.

Åström, K.J., and B. Wittenmark (1973). On self-tuning regulators, *Automatica*, **9**, 185-199.

Åström, K.J., and B. Wittenmark (1984). *Computer-Controlled Systems, Theory and Design*, Prentice-Hall, Englewood Cliffs.

Atkinson, A.C. (1978). Posterior probabilities for choosing a regression model, *Biometrika*, **65**(1), 39-48.

Atkinson, A.C. (1985). *Plots, Transformations and Regression*, Oxford University Press, Oxford.

Atkinson, A.C. (1992). Optimum experimental designs for parameter estimation and for discrimination between models in the presence of prior information, in *Model-Oriented Data Analysis, A survey of Recent Methods* (V.V. Fedorov, W.G. Müller and I.N. Vuchkov, eds.), Physica-Verlag, Heidelberg, 3-30.

Atkinson, A.C. (1995). Multivariate transformations, regression diagnostics and seemingly unrelated regression, in *MODA4, Advances in Model-Oriented Data Analysis* (Ch.P. Kitsos and W.G. Müller, eds.), Physica-Verlag, Heidelberg, 181-192.

Atkinson, A.C., K. Chaloner, A.M. Hertzberg and J. Juritz (1993). Optimum experimental designs for properties of a compartmental model, *Biometrics*, **49**, 325-337.

Atkinson, A.C., and D.R. Cox (1974). Planning experiments for discriminating between models (with discussion), *J. Royal Statist. Soc.*, **B36**, 321-348.

Atkinson, A.C., and A.N. Donev (1989). The construction of exact D-optimum experimental designs with application to blocking response surface designs, *Biometrika*, **76**(3), 515-526.

Atkinson, A.C., and A.N. Donev (1992). *Optimum Experimental Design*, Oxford University Press.

Atkinson, A.C., and V.V. Fedorov (1975a). The design of experiments for discriminating between two rival models, *Biometrika*, **62**(1), 57-70.

Atkinson, A.C., and V.V. Fedorov (1975b). Optimal design: experiments for discriminating between several rival models, *Biometrika*, **62**(2), 289-303.

Atkinson, A.C., and W.G. Hunter (1968). The design of experiments for parameter estimation, *Technometrics*, **10**(2), 271-289.

Atwood, C.L. (1973). Sequences converging to D-optimal designs of experiments, *Annals of Stat.*, **1**(2), 342-352.

Atwood, C.L. (1976). Convergent design sequences for sufficiently regular optimality criteria, *Annals of Stat.*, **4**(6), 1124-1138.

Atwood, C.L. (1980). Convergent design sequences for sufficiently regular optimality criteria, II: singular case, *Annals of Stat.*, **8**(4), 894-912.

Bandemer, H., W. Näther and J. Pilz (1987). Once more: optimal experimental design for regression models (with discussion), *Statistics*, **19**(4), 171-217.

Bard, J. (1974). *Nonlinear Parameter Estimation*, Academic Press, New York.

Barham, R.H., and W. Drane (1972). An algorithm for least squares estimation of nonlinear parameters when some of the parameters are linear, *Technometrics*, **14**, 757-766.

Bar-Shalom, Y., and T.E. Fortmann (1988). *Tracking and Data Association*, Academic Press, New York.

Bates, D.M., and D.G. Watts (1980). Relative curvature measures of nonlinearity, *J. Royal Statist. Soc.*, **B42**, 1-25.

Bates, D.M., and D.G. Watts (1981). Parameter transformation for improved approximate confidence regions in nonlinear least squares, *Annals of Stat.*, **9**(6), 1152-1167.

Bates, D.M., and D.G. Watts (1988). *Nonlinear Regression Analysis and its Applications*, Wiley, New York.

Bayard, D., and A. Schumitzky (1990). A stochastic control approach to optimal sampling design, *Technical Report* 90-1, University of Southern California, School of Medicine, Lab. of Applied Pharmacokinetics.

Bayard, D., M.H. Milman and A. Schumitzky (1994). Design of dosage regimens: a multiple model stochastic control approach, *Int. J. Bio-medical Computing*, **36**, 103-115.

Beghelli, S., R.P. Guidorzi and U. Soverini (1990). The Frisch scheme in dynamic system identification, *Automatica*, **26**, 171-176.

Bekey, G.A., and S.F. Masri (1983). Random search techniques for optimization of nonlinear systems with many parameters, *Math. and Comput. in Simulation*, **25**, 210-213.

Belforte, G., B. Bona and V. Cerone (1990). Parameter estimation algorithms for a set-membership description of uncertainty, *Automatica*, **26**, 887-898.

Bellman, R., and K. J. Åström (1970). On structural identifiability. *Math. Biosci.*, **7**, 329-339.

Benveniste, A., M. Métivier and P. Priouret (1987). *Algorithmes adaptatifs et approximations stochastiques, théorie et applications*, Masson, Paris.

Benveniste, A., M. Métivier and P. Priouret (1990). *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, Berlin.

Berger, M. (1979). *Géométrie*, CEDIC/Nathan, Paris.

Berger, M. (1987). *Geometry I and II*, Springer-Verlag, Berlin.

Berman, M., and R. Schoenfeld (1956). Invariants in experimental data on linear kinetics and the formulation of models. *J. of Applied Physics*, **27**, 1361-1370.

Berthier, F., J.-P. Diard, C. Montella, L. Pronzato and É. Walter (1995), Choice of experimental method induced by structural properties of mechanisms, *J. of Electroanal. Chemistry*, **399**, 1-6.

Berthier, F., J.-P. Diard, C. Montella, L. Pronzato and É. Walter (1996), Identifiability and distinguishability concepts in electrochemistry, *Automatica*, **32**(7), 973-984.

Bierman, G.J. (1977). *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York.

Bilardello, P., X. Joulia, J.M. Le Lann, H. Delmas and B. Koehret (1993). A general strategy for parameter estimation in differential-algebraic systems, *Computers Chem. Engng.*, **17**, 517-525.

Bitmead, R.R., M. Gevers and V. Wertz (1990). *Adaptive Optimal Control, the Thinking Man's GPC*, Prentice-Hall, Englewood Cliffs.

Bland, R.G., O. Goldfarb and M.J. Todd (1981). The ellipsoid method: a survey, *Operations Research*, **29**(6), 1039-1051.

Blight, B.J.N., and L. Ott (1975). A Bayesian approach to model inadequacy for polynomial regression, *Biometrika*, **62**(1), 79-88.

Bloomfield, P., and W.L. Steiger (1983). *Least Absolute Deviations: Theory, Applications and Algorithms*, Birkhäuser, Boston.

Bohachevsky, I.O. , M.E. Johnson and M.L. Stein (1986). Generalized simulated annealing for function optimization, *Technometrics*, **28**(3), 209-217.

Böhning, D. (1985). Numerical estimation of a probability measure, *J. Statist. Planning and Inference*, **11**, 57-69.

Böhning, D. (1989). Likelihood inference for mixtures: geometrical and other constructions of monotone step-length algorithms, *Biometrika*, **76**(2), 375-383.

Bonnans, J.-F. (1987). Théorie de la pénalisation exacte, *Rapport de Recherche INRIA* 743.

Borrie, J.A. (1992). *Stochastic Systems for Engineers*, Prentice-Hall, Hemel Hempstead.

Borth, D.M. (1975). A total entropy criterion for the dual problem of model discrimination and parameter estimation, *J. Royal Statist. Soc.*, **B37**(1), 77-87.

Box, M.J. (1968). The occurence of replications in optimal design of experiments to estimate parameters in nonlinear models, *J. Royal Statist. Soc.*, **30**, 290-302.

Box, M.J. (1970). Some experiences with a nonlinear experimental design criterion, *Technometrics*, **12**(3), 569-589.

Box, M.J. (1971). Bias in nonlinear estimation, *J. Royal Statist. Soc.*, **B33**, 171-201.

Box, G.E.P., and D.R. Cox (1964). An analysis of transformations (with discussion), *J. Royal Statist. Soc.*, **B26**, 211-246.

Box, G.E.P., and N.R. Draper (1959). A basis for the selection of a response surface design, *J. Am. Statist. Ass.*, **54**, 622-654.

Box, G.E.P., and N.R. Draper (1987). *Empirical Model-Building and Response Surfaces*, Wiley, New York.

Box, G.E.P., and W.J. Hill (1967). Discriminating between mechanistic models, *Technometrics*, **9**(1), 57-71.

Box, G.E.P., and W.J. Hill (1974). Correcting inhomogeneity of variance with power transformation weighting, *Technometrics*, **16**, 385-389.

Box, G.E.P., and J.S. Hunter (1957). Multi-factor experimental designs for exploring response surfaces, *Annals Math. Stat.*, **28**, 195-241.

Box, G.E.P., and W.G. Hunter (1965). The experimental study of physical mechanisms, *Technometrics*, **7**(1), 23-42.

Box, G.E.P., and G.M. Jenkins (1976). *Time Series Analysis Forecasting and Control*, Holden Day, San Francisco.

Box, G.E.P., and H.L. Lucas (1959). Design of experiments in nonlinear situations, *Biometrika*, **46**, 77-90.

Box, G.E.P., and G.C. Tiao (1973). *Bayesian Inference in Statistical Analysis*, Addison-Wesley, Reading.

Box, G.E.P., and K.B. Wilson (1951). On the experimental attainment of optimum conditions (with discussion), *J. Royal Statist. Soc.*, **B13**(1), 1-45.

Boyles, R.A. (1983). On the convergence of the EM algorithm, *J. Royal Statist. Soc.*, **B45**(1), 47-50.

Bozin, A., and M. Zarrop (1991). Self tuning optimizer — convergence and robustness properties, in *Proc. 1st European Control Conf.*, Grenoble, 672-677.

Brent, R.P. (1973). *Algorithms for Minimization Without Derivatives*, Prentice-Hall, Englewood Cliffs.

Breusch, T.S., and A.R. Pagan (1979). A simple test for heteroscedasticity and random coefficient variation. *Econometrica*, **47**, 1287-1294.

Broman, V., and M.J. Shensa (1990). A compact algorithm for the intersection and approximation of N-dimensional polytopes, *Math. and Comput. in Simulation*, **32**, 469-480.

Brooks, R.J. (1977). Optimal regression design for control in linear regression, *Biometrika*, **64**(2), 319-325.

Brooks, S.H., and R.M. Mickey (1961). Optimum estimation of gradient direction in steepest ascent experiments, *Biometrics*, **17**, 48-56.

Buchberger, B. (1970). Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. *Aeq. Math.*, **4**, 374-383.

Caines, P.E. (1988). *Linear Stochastic Systems*. Wiley, New York.

Carrillo Le Roux, G. (1995). *Stratégie d'identification de modèles algébro-différentiels, application aux systèmes réactionnels complexes*, Thèse de Doctorat en Sciences, Institut National Polytechnique de Toulouse.

Cerone, V. (1991). Parameter bounds for models with bounded errors in all variables, in *Prep. 9th IFAC/IFORS Symp. Identification and System Parameter Estimation*, Budapest, 1518-1523.

Cerone, V. (1993). Feasible parameter set for linear models with bounded errors in all variables, *Automatica*, **29**, 1551-1555.

Chalam, V.V. (1987). *Adaptive Control Systems, Techniques and Applications*, Marcel Dekker.

Chaloner, K. (1984). Optimal Bayesian experimental design for linear models, *The Annals of Stat.*, **12**(1), 283-300.

Chaloner, K., and K. Larntz (1986). Optimal Bayesian design applied to logistic regression experiments, *Technical Report* 483, University of Minnesota.

Chaloner, K., and K. Larntz (1988). Software for logistic regression experiment design, in *Optimal Design and Analysis of Experiments* (Y. Dodge, V.V. Fedorov and H.P. Wynn, eds.), North-Holland, Amsterdam, 207-211.

Chaloner, K., and I. Verdinelli (1995). Bayesian experimental design: a review, *Statistical Science*, **10**(3), 273-304.

Chappell, M. J., K. R. Godfrey and S. Vajda (1990). Global identifiability of the parameters of nonlinear systems with specified inputs: a comparison of methods. *Math. Biosci.*, **102**, 41-73.

Chatterjee, S.K., and N.K. Mandal (1981). Response surface designs for estimating the optimal point, *Calcutta Statist. Assoc. Bull.*, **30**, 145-169.

Chatterjee, S.K., and N.K. Mandal (1985). Designs for hitting the bull's eye of a response surface, in *Biostatistics: Statistics in Biomedical, Public Health and Environmental Sciences* (P.K. Sen, ed.), Elsevier, North-Holland, Amsterdam, 283-303.

Chen, S., and S.A. Billings (1989). Representation of non-linear systems: the NARMAX model, *Int. J. Control*, **49**(3), 1013-1032.

Chernoff, H. (1953). Locally optimal designs for estimating parameters, *Annals of Math. Stat.*, **24**, 586-602.

Chernoff, H. (1975). Approaches in sequential design of experiments, in *Survey of Statistical Design and Linear Models* (J.N. Srivastava, ed.), North-Holland, Amsterdam, 67-90.

Chernov, N.I., G.A. Ososkov and L. Pronzato (1992). Three-dimensional multivertex reconstruction from two-dimensional tracks observations using likelihood inference, *Applications of Math.* (Prague), **37**(6), 437-452.

Cheung, M.-F., S. Yurkovitch and K.M. Passino (1993). An optimal volume ellipsoid algorithm for parameter set estimation, *IEEE Trans. on Autom. Control*, **38**, 1292-1296.

Clarke, D.W. (1967). Generalized least squares estimation of the parameters of a dynamic model, in *Proc. 1st IFAC Symp. Identification in Automatic Control Systems*, Prague, Paper 3.17.

Clarke, D.W. (1988). Application of generalized predictive control to industrial processes, *IEEE Control Systems Magazine*, **4**, 49-55.

Clarke, D.W., C. Mohtadi and P.S. Tuffs (1987). Generalized predictive control, Parts I and II, *Automatica*, **23**, 137-160.

Clément T., and S. Gentil (1988). Reformulation of parameter identification with unknown-but-bounded errors, *Math. and Comput. in Simulation*, **30**, 257-270.

Clément T., and S. Gentil (1990). Recursive membership set estimation for output-error models, *Math. and Comput. in Simulation*, **32**, 505-513.

Cobelli, C., A. Ruggeri and K. Thomaseth (1984). Optimal input and sampling schedule design for physiological system identification. Some theoretical and in vivo results on a compartmental model of glucose kinetics, in *Mathematics and Computers in Biomedical applications* (J. Eisenfeld and C. DeLisi, eds.), Elsevier, North-Holland, Amsterdam.

Cobelli, C., and K. Thomaseth (1985). Optimal input design for identification of compartmental models. Theory and application to a model of glucose kinetics, *Math. Biosci.*, **77**, 267-286.

Cobelli, C., and K. Thomaseth (1988a). On optimality of the impulse input for linear system identification, *Math. Biosci.*, **89**, 127-133.

Cobelli, C., and K. Thomaseth (1988b). Optimal equidose inputs and role of measurement error for estimating the parameters of a compartmental model of glucose kinetics from continuous- and discrete-time optimal samples, *Math. Biosci.*, **89**, 135-147.

Cochran, W.G., and G.M. Cox (1957). *Experimental Designs* (2nd edition), Wiley, New York.

Combettes, P.L. (1993). The foundations of set theoretic estimation, *Proc. of the IEEE*, **81**(2), 182-208.

Cook, D.R., and C.J. Nachtsheim (1980). A comparison of algorithms for constructing exact D-optimal designs, *Technometrics*, **22**(3), 315-324.

Cook, D.R., and S. Weisberg (1982). *Residuals and Influence in Regression*, Chapman and Hall, New York.

Cox, D.R. (1958). *Planning of Experiments*, Wiley, New York.

Currin, C., T. Mitchell, M. Morris and D. Ylvisaker (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments, *J. Am. Statist. Ass.*, **86**, 953-963.

Dang Van Mien, H. (1973). Utilisation pratique de l'algorithme de B.L. Ho. Application à la réduction des systèmes multidimensionnels. Amélioration par la méthode des moindres carrés, *Bull. E.D.F. Études et Recherches*, Série C. Math. Info., **2**, 23-46.

Dang Van Mien, H., and D. Normand-Cyrot (1984). Nonlinear state affine identification methods: applications to electrical power plants, *Automatica*, **20**(2), 175-188.

Dantzig, G.B. (1963). *Linear Programming and Extensions*, Princeton University Press, Princeton, N.J.

D'Argenio, D.Z. (1981). Optimal sampling times for pharmacokinetic experiments, *J. Pharm. Biopharm.*, **9**(6), 739-756.

D'Argenio, D.Z. (1990). Incorporating prior parameter uncertainty in the design of sampling schedules for pharmacokinetic parameter estimation experiments, *Math. Biosci.*, **99**, 105-118.

D'Argenio, D.Z., and M. Van Guilder (1988). Design of experiments for parameter estimation involving uncertain systems, with application to pharmacokinetics, in *Prep. 12th IMACS World Congress on Scientific Computation*, Paris, **2**, 511-513.

Dasgupta, A. (1991). Diameter and volume minimizing confidence sets in Bayes and classical problems, *The Annals of Stat.*, **19**(3), 1225-1243.

Davenport, J., Y. Siret and E. Tournier (1987). *Calcul formel*, Masson, Paris.

Davenport, J., Y. Siret and E. Tournier (1993). *Computer Algebra*, Academic Press, London.

Deller, J.R., M. Nayeri and S.H. Odeh (1993). Least-square identification with error bounds for real-time signal processing and control, *Proc. of the IEEE*, **81**(6), 815-849.

De Moor, B. (1988). *Mathematical Concepts and Techniques for Modelling of Static and Dynamic Systems*, Ph.D. dissertation, Katholieke Universiteit Leuven.

De Moor, B., and J. Vandewalle (1986a). A geometric approach to the maximal corank problem in the analysis of linear relations, in *Proc. 25th Conf. on Decision and Control*, Athens, 1990-1995.

De Moor, B., and J. Vandewalle (1986b). The uniqueness versus the non-uniqueness principle in the identification of linear relations from noisy data, in *Proc. 25th Conf. on Decision and Control*, Athens, 1663-1665.

De Moor, B., and P. Van Overschee (1995). Numerical algorithms for subspace state space system identification, in *Trends in Control, a European Perspective* (A. Isidori, ed.), Springer-Verlag, Berlin.

Demoment, G. (1989). Image reconstruction and restoration: overview of common estimation structures and problems, *IEEE Trans. Acoustics, Speach and Signal Processing*, **37**(12), 2024-2036.

Dempster, A.P., N. Laird and D.B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm, *J. Royal Statist. Soc.*, **B39**, 1-38.

den Hertog, D. (1994). *Interior Point Approach to Linear, Quadratic and Convex Programming*, Kluwer, Dordrecht.

Diaconis, P., and B. Efron (1983). Méthodes de calculs statistiques intensifs sur ordinateurs, *Pour la Science*, juillet, 46-58.

DiCiccio, T.J., and J.P. Romano (1988). A review of bootstrap confidence intervals (with discussion), *J. Royal Statist. Soc.*, **B50**(3), 338-370.

Didrit, O., L. Jaulin and É. Walter (1995). Guaranteed analysis and optimization of parametric systems, with application to their stability degree, in *Proc. 3rd European Control Conf.*, Rome, 1412-1417.

Diop, S., and M. Fliess (1991). Nonlinear observability, identifiability and persistent trajectories, in *Proc. 30th IEEE Conf. on Decision and Control*, Brighton, 714-719.

Dixon, L.C.W., and G.P. Szegö (1975). *Towards Global Optimisation 1*, North-Holland, Amsterdam.

Dixon, L.C.W., and G.P. Szegö (1978). *Towards Global Optimisation 2*, North-Holland, Amsterdam.

Djuric, P.M., and S.M. Kay (1994). Model selection based on Bayesian predictive densities and multiple data records, *IEEE Trans. Signal Processing*, **42**(7), 1685-1699.

Dodge, Y. (ed.) (1987). *Statistical Data Analysis Based on the L_1-Norm and Related Methods*, North-Holland, Amsterdam.

Dodge, Y., V.V. Fedorov and H.P. Wynn (eds.) (1988). *Optimal Design and Analysis of Experiments*, North-Holland, Amsterdam.

Draper, N.R. and H. Smith (1981). *Applied Regression Analysis*, Wiley, New York.

Durieu, C., B.T. Polyak and É. Walter (1996a), Trace versus determinant in ellipsoidal outer-bounding, with application to state estimation, in *Prep. 13th IFAC World Congress*, San Francisco, **I**, 43-48.

Durieu, C., B.T. Polyak and É. Walter (1996b), Ellipsoidal state outer-bounding for MIMO systems via analytical techniques, in *Proc. CESA 1996 IMACS IEEE-SMC Multiconference*, Lille, Symposium on Modelling, Analysis and Simulation, **2**, 843-848.

Dvoretzky, A. (1956). On stochastic approximation, in *Proc. 3rd Berkeley Symp. Math. Stat. and Probability*, 39-55.

Eadie, W.T., D. Drijard, F.E. James, M. Roos and B. Sadoulet (1971). *Statistical Methods in Experimental Physics*, North-Holland, Amsterdam.

Ecker, J.G., and M. Kupferschmid (1985). A computational comparison of the ellipsoidal algorithm with several nonlinear programming algorithms, *SIAM J. Control and Optimization*, **23**(5), 657-674.

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans*, SIAM CBMS-NSF monograph **38**, Philadelphia.

Ermakov, S.M. (ed.) (1983). *Mathematical Theory of Experiment Design*, Nauka, Moscow (in Russian).

Ermakov, S.M., and A.A. Zhigljavsky (1987). *Mathematical Theory of Optimal Experiments*, Nauka, Moscow (in Russian).

Ermoliev, Yu., and R.J-B. Wets (eds.) (1988). *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin.

Eykhoff, P. (1974). *System Identification, Parameter and State Estimation*, Wiley, London.

Farebrother, R.W. (1987). The historical development of the L_1 and L_∞ estimation procedures 1793-1930, in *Statistical Data Analysis Based on the L_1-norm and Related Methods*, (Y. Dodge, ed.), North-Holland, Amsterdam, 37-63.

Fedorov, V.V. (1972). *Theory of Optimal Experiments*, Academic Press, New York.

Fedorov, V.V. (1975). Optimal experimental designs for discriminating two rival regression models, in *A Survey of Statistical Design and Linear Models* (J.N. Srivastava, ed.), North-Holland, Amsterdam, 155-164.

Fedorov, V.V. (1980). Convex design theory, *Math. Operationsforsch. Statist., Ser. Statistics*, **11**(3), 403-413.

Fedorov, V.V., and A.C. Atkinson (1988). The optimum design of experiments in the presence of uncontrolled variability and prior information, in *Optimal Design and Analysis of Experiments* (Y. Dodge, V.V. Fedorov and H.P. Wynn, eds.), North-Holland, Amsterdam, 327-344.

Fedorov, V.V., and A. Pázman (1968). Design of physical experiments (statistical methods), *Fortschritte der Physik*, **16**(6), 325-355.

Filippova, T.F., A.B. Kurzhanski, K. Sugimoto and I. Vályi (1996). Ellipsoidal state estimation for uncertain dynamical systems, in *Bounding Approaches to System Identification* (M. Milanese *et al.*, eds.), Plenum Press, New York, 213-238.

Finney, D.J. (1960). *An Introduction to the Theory of Experimental Design*, University of Chicago Press.

Firth, D. (1993). Bias reduction of maximum likelihood estimates, *Biometrika*, **80**(1), 27-38.

Fisher, F. M. (1966). *The Identification Problem in Economics*. McGraw-Hill, New York.

Fisher, R.A. (1925). *Statistical Methods for Research Workers*, Oliver and Boyd, Edinburgh.

Fisher, R.A. (1926). The arrangement of field experiments, *J. of the Ministry of Agriculture*, **33**, 503-513.

Fletcher, R., and M.J.D. Powell (1963). A rapidly convergent descent method for minimization, *Computer J.*, June, 163-168.

Fletcher, R., and M.C. Reeves (1964). Function minimization by conjugate gradients, *Computer J.*, July, 149-154.

Fliess, M. (1978). Un codage non commutatif pour certains systèmes échantillonnés non linéaires, *Information and Control*, **38**(3), 264-287.

Fliess, M. (1989). Automatique et corps différentiels. *Forum Math.*, **1**, 227-238.

Fogel, E., and Y.F. Huang (1982). On the value of information in system identification — bounded noise case, *Automatica*, **18**, 229-238.

Ford, I., Ch.P. Kitsos and D.M. Titterington (1989). Recent advances in nonlinear experimental design, *Technometrics*, **31**(1), 49-60.

Ford, I., and S.D. Silvey (1980). A sequentially constructed design for estimating a nonlinear parametric function, *Biometrika*, **67**(2), 381-388.

Ford, I., D.M. Titterington and C.F.J. Wu (1985). Inference and sequential design, *Biometrika*, **72**(3), 545-551.

Forsythe, G.E. (1968). On the asymptotic directions of the s-dimensional optimum gradient method, *Numerische Mathematik*, **11**, 57-76.

Fourgeaud, C., and A. Fuchs (1967). *Statistique*, Dunod, Paris.

Frisch, R. (1934). *Statistical Confluence Analysis by Means of Complete Regression Systems*, Pub. No. 5, Economic Institute, Oslo University.

Galil, Z., and J. Kiefer (1977). Comparison of simplex designs for quadratic mixture models, *Technometrics*, **19**(3), 445-453.

Galil, Z., and J. Kiefer (1980). Time- and space-saving computer methods, related to Mitchell's DETMAX, for finding D-optimum designs, *Technometrics*, **22**(3), 301-313.

Gelfand, A., S.E. Hills, A. Racine-Poon and A.F. Smith (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling, *J. Am. Stat. Assoc.*, **85**, 972-985.

Genz, A.C., and A.A. Malik (1980). Remarks on algorithm 006: an adaptive algorithm for numerical integration over an N-dimensional rectangular region, *J. Computational and Applied Math.*, **6**(4), 295-302.

Gevers, M. (1993). Towards a joint design of identification and control?, in *Essays on Control: Perspectives in the Theory and its Applications* (H.L. Trentelman and J.C. Willems, eds.), Birkhäuser, Boston, 111-151.

Gevers, M., and L. Ljung (1985). Benefits of feedback in experiment design, in *Prep. 7th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, York, 909-914.

Gevers, M., and L. Ljung (1986). Optimal experiment design with respect to the intended model application, *Automatica*, **22**, 543-554.

Gilbert, J.C., G. Le Vey and J. Masse (1991). La différentiation automatique de fonctions représentées par des programmes. *Rapport de Recherche INRIA* 1557, Rocquencourt.

Gill, P.E., and W. Murray (eds.) (1974). *Numerical Methods for Constrained Optimization*, Academic Press, New York.

Gill, P.E., W. Murray and M.H. Wright (1981). *Practical Optimization*, Academic Press, London.

Glover, K., and J. C. Willems (1974) Parametrizations of linear dynamical systems: canonical forms and identifiability. *IEEE Trans. Autom. Control*, **AC-19**, 640-644.

Goldstein, M. (1980). The linear Bayes regression estimator under weak prior assumptions, *Biometrika*, **67**(3), 621-628.

Goldstein, M., and A.F.M. Smith (1974). Ridge-type estimators for regression analysis, *J. Royal Statist. Soc.*, **B36**, 284-291.

Golub, G.H., M. Heath and G. Wahba (1979). Generalized cross-validation as a method for choosing a good ridge parameter, *Technometrics*, **21**, 215-223.

Golub, G.H., and V. Pereyra (1973). The differentiation of pseudo-inverses and non-linear least squares problems whose variables separate, *J. of SIAM*, **10**, 413-432.

Gonin, R., and A.H. Money (1989). *Nonlinear L_p-norm Estimation*, Marcel Dekker, New York.

Gonzaga, C.C. (1992). Path-following methods for linear programming, *SIAM Review*, **34**(2), 167-224.

Goodwin, G.C., J.C. Murdoch and R.L. Payne (1973). Optimal test signal design for linear SISO system identification, *Int. J. Control*, **17**(1), 45-55.

Goodwin, G.C., and R.L. Payne (1977). *Dynamic System Identification: Experiment Design and Data Analysis*, Academic Press, New York.

Goodwin, G.C., and K.S. Sin (1984). *Adaptive Filtering, Prediction and Control*, Prentice-Hall, Englewood Cliffs.

Goodwin, G.C., M.B. Zarrop and R.L. Payne (1974). Coupled design of test signals, sampling intervals, and filters for system identification, *IEEE Trans. Autom. Control*, **19**, 748-752.

Gorman, J.D., and A.O. Hero (1990). Lower bounds for parametric estimation with constraints, *IEEE Trans. on Information Theory*, **36**(6), 1285-1301.

Grant, F.H., and J.J. Solberg (1983). Variance reduction techniques in stochastic shortest route analysis: application procedure and results, *Math. and Comput. in Simulation*, **25**, 366-375.

Grewal, M.S., and K. Glover (1975). Relationships between identifiability and input selection, in *Proc. 14th IEEE Conf. on Decision and Control*, 526-528.

Grewal, M. S., and K. Glover (1976). Identifiability of linear and nonlinear dynamical systems. *IEEE Trans. Autom. Control*, **AC-21**, 833-837.

Griewank, A., and G. Corliss (eds.) (1991). *Automatic Differentiation of Algorithms: Theory, Implementation and Application*, SIAM, Philadelphia.

Grötschel, M., L. Lovasz and A. Schrijver (1988). *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin.

Guidorzi, R.P. (1991). Certain models from uncertain data: the algebraic case, *Systems and Control Letters*, **17**, 415-424.

Gustavsson, I., L. Ljung and T. Söderström (1981). Choice and effect of different feedback configurations, in *Trends and Progress in System Identification* (P. Eykhoff, ed.), Pergamon, Oxford, 367-388.

Halperin, M. (1963). Confidence interval estimation in nonlinear regression, *J. Royal Statist. Soc.*, **B25**, 330-333.

Hammer, R., M. Hocks, U. Kulisch and D. Ratz (1995), *C^{++} Toolbox for Verified Computing*, Springer, Berlin.

Hamilton, D.C., and D.G. Watts (1985). A quadratic design criterion for precise estimation in nonlinear regression models, *Technometrics*, **27**, 241-250.

Hamilton, D.C., D.G. Watts and D.M. Bates (1982). Accounting for intrinsic nonlinearities in nonlinear regression parameter inference regions, *The Annals of Stat.*, **10**, 386-393.

Hansen, E. (1992). *Global Optimization Using Interval Analysis*, Marcel Dekker, New York.

Happel, J. (1986). *Isotopic Assessment of Heterogeneous Catalysis*. Academic Press, Orlando.

Hasting-James, R., and M.W. Sage (1969). Recursive generalized-least-squares procedure for on-line identification of process parameters, *Proc. IEE*, **116**, 2057-2062.

Herbin, H., A. Venot, J.-Y. Devaux, É. Walter, J.-F. Lebruchec, L. Dubertret and J.-C. Roucayrol (1989). Automated registration of dissimilar images: application to medical imagery, *Computer Vision, Graphics and Image Processing*, **47**, 77-88.

Hermann, R., and A. J. Krener (1977). Nonlinear controllability and observability, *IEEE Trans. Autom. Control*, **AC-22**, 728-740.

Heuberger, P.S.C., P.M.J. Van den Hof and O.H. Bosgra (1995). A generalized orthonormal basis for linear dynamical systems, *IEEE Trans. Autom. Control*, **40**, 451-465.

Hill, P.D.H. (1978). A review of experimental design procedures for regression model discrimination, *Technometrics*, **20**, 15-21.

Hill, W.J., and W.G. Hunter (1966). A review of response surface methodology: a literature survey, *Technometrics*, **8**, 571-590.

Hill, W.J., W.G. Hunter and D.W. Wichern (1968). A joint design criterion for the dual problem of model discrimination and parameter estimation, *Technometrics*, **10**(1), 145-160.

Hindmarsh, A.C. (1980). LSODE and LSODI, two initial value ordinary differential equation solvers, *ACM SIGNUM Newsletter*, **15**, 10-11.

Hinkley, D.V. (1988). Bootstrap methods, *J. Royal Statist. Soc*, **B50**(3), 321-337.

Hiriart-Urruty, J.B., and C. Lemaréchal (1993). *Convex Analysis and Minimization Algorithms*, Parts 1 and 2, Springer-Verlag, Berlin.

Hjalmarsson, H., M. Gevers and F. De Bruyne (1996). For model-based control design, closed-loop identification gives better performances, *Automatica*, to appear.

Hjalmarsson, H., and L. Ljung (1994). A unifying view of disturbances in identification, in *Prep. 10th IFAC/IFORS Symp. on System Identification* (M. Blanke and T. Söderström, eds.), Copenhagen, 2, 73-78.

Ho, B.L., and R.E. Kalman (1966). Effective construction of linear state-variable models from input-output functions, *Regelungstechnik*, **14**, 545-548.

Hoerl, A.E., and R.W. Kennard (1970). Ridge regression: biased estimation for nonorthogonal problems, *Technometrics*, **12**, 55-68.

Hornik, K., M. Stinchcombe and H. White (1989). Multilayer feedforward networks are universal approximators, *Neural Networks*, **2**, 359-366.

Horst, R., and H. Tuy (1990). *Global Optimization, Deterministic Approaches*, Springer-Verlag, Berlin.

Hougaard, P. (1985). Saddlepoint approximations for curved exponential families, *Statist. Prob. Lett.*, **3**, 161-166.

Huang, C.-Y. (1991). *Planification d'expériences pour la discrimination entre structures de modèles*. Thèse de Doctorat en Sciences, Université de Paris-Sud, Orsay.

Huang, C.-Y., L. Pronzato and É. Walter (1991). Nonsequential T-optimal design for model discrimination: new algorithms, in *Proc. Probastat'91*, Bratislava, (A. Pázman and J. Volaufová, eds.), Slovak Ak. Sciences, 130-136.

Huber, P.J. (1975). Robustness and designs, in *A survey of Statistical Design and Linear Models* (J.N. Srivastava, ed.), North-Holland, Amsterdam, 287-301.

Huber, P.J. (1981). *Robust Statistics*, Wiley, New York.

IBM (1986). *High Accuracy Arithmetic Subroutine Library* (ACRITH), Program description and user's guide, SC 33-6164-02, third edition.

Irving, E., J. Daoudi and H. Bourlès (1991). The combined robust adaptive control, in *Proc. 1st European Control Conf.*, Grenoble, 252-257.

Isermann, R. (1974). *Prozeßidentifikation, identifikation und Paramterschätzung dynamischer Prozesse mit diskreten Signalen*, Springer-Verlag, Berlin.

Isermann, R., K.-H. Lachmann and D. Matko (1992). *Adaptive Control Systems*, Prentice-Hall, Hemel Hempstead.

Jackson, J.E. (1991). *A User's Guide to Principal Components*, Wiley, New York.

Jaulin, L., and É. Walter (1993a). Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis, *Math. and Comput. in Simulation*, **35**, 1923-1937.

Jaulin, L., and É. Walter (1993b). Set-inversion via interval analysis for nonlinear bounded-error estimation, *Automatica*, **29**, 1053-1064.

Jaulin, L., É. Walter and O. Didrit (1996). Guaranteed Robust nonlinear parameter bounding, in *Proc. CESA 1996 IMACS IEEE-SMC Multiconference*, Lille, Symposium on Modelling, Analysis and Simulation, **2**, 1156-1161.

Jazwinski, A.H. (1970). *Stochastic Processes and Filtering Theory*, Academic Press, New York.

Jelliffe, R.W., and A. Schumitzky (1990). Modeling, adaptive control, and optimal drug therapy, *Medical Progress through Technology*, **16**, 85-110.

Jennrich, R.I., and M.L. Ralston (1979). Fitting nonlinear models to data, *Ann. Rev. Biophys. Bioeng.*, **8**, 195-238.

Johnson, M.E., and C.J. Nachtsheim (1983). Some guidelines for constructing exact D-optimal designs on convex design spaces, *Technometrics*, **25**(3), 271-277.

Kaczmarz, S. (1937). Angenäherte Auflösung von Systemen Linearer Gleichungen, *Bull. Int. Acad. Polon., Sci. Mat. Nat., Ser. A*.

Kalaba, R.E., and K. Spingarn (1974). Optimal inputs for nonlinear process parameter estimation, *IEEE Trans. Aerospace and Electronic Systems*, **10**, 339-345.

Kalaba, R.E., and K. Spingarn (1981). Optimal inputs for blood glucose regulation parameter estimation, part 3, *J. Optimization Theory and Applications*, **33**, 267-285.

Kalaba, R.E., and K. Spingarn (1982). *Control, Identification and Input Optimization*, Plenum Press, New York.

Kalaba, R.E., and K. Spingarn (1984). Automatic solution of optimal control problems III: differential and integral constraints, *Control Systems Magazine*, February, 3-8.

Kalman, R.E. (1982). Identification from real data, in *Current Developments in the Interface: Economics, Econometrics, Mathematics* (M. Hazewinkel and H.G. Rinnooy Kan, eds.), Reidel, Dortrecht, 161-196.

Kanji, G.K. (1993). *100 Statistical Tests*. Sage Publications, London.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming, *Combinatorica*, **4**(4), 373-395.

Karson, M.J., A.R. Manson and R.J. Hader (1969). Minimum bias estimation and experimental design for response surfaces, *Technometrics*, **11**, 461-475.

Kaufman, H., I. Bar-Kana and K. Sobel (1994). *Direct Adaptive Control Algorithms, Theory and Applications*, Springer-Verlag, London.

Kearfott, R.B., and V. Kreinovich (eds.) (1996). *Applications of Interval Computations*, Kluwer, Dordrecht.

Keating, P., R.L. Mason and P.K. Sen (1993). *Pitman's Measure of Closeness: a Comparison of Statistical Estimators*, SIAM, Philadelphia.

Kelley, J.E. (1960). The cutting plane method for solving convex programs, *SIAM J.*, **8**, 703-712.

Kendall, M.G., and A. Stuart (1967). *The Advanced Theory of Statistics*, **2**, *Inference and Relationship*, second edition, Charles Griffin and Company, London.

Keviczky, L. (1975). "Design of experiments" for the identification of linear dynamic systems, *Technometrics*, **17**(3), 303-308.

Khachiyan, L.G. (1979). A polynomial algorithm in linear programming, *Doklady AkademiïaNauk SSSR*, **244**, 1093-1096 (translated in English in *Soviet Mathematics Doklady*, **20**, 191-194).

Khachiyan, L.G., and M.J. Todd (1993). On the complexity of approximating the maximal inscribed ellipsoid for a polytope, *Mathematical Programming*, **A61**(2), 137-159.

Kiefer, J. (1953). Sequential minimax search for a maximum, *Proc. Am. Math. Soc.*, **4**, 502-506.

Kiefer, J. (1956). Optimum sequential search and approximation methods under minimum regularity assumptions, *J. SIAM*, **5**(3), 105-136.

Kiefer, J. (1961). Optimum designs in regression problems II, *Annals Math. Stat.*, **32**, 298-325.

Kiefer, J. (1973). Optimum designs for fitting biased multiresponse surfaces, in *Multivariate Analysis 3* (P.R. Krishnaiah, ed.), Academic Press, New York, 287-397.

Kiefer, J. (1974). General equivalence theory for optimum designs (approximate theory), *Annals of Stat.*, **2**(5), 849-879.

Kiefer, J., and J. Wolfowitz (1952). Stochastic estimation of the maximum of a regression function, *Annals of Math. Stat.*, **23**, 462-466.

Kiefer, J., and J. Wolfowitz (1956). Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters, *Annals of Math. Stat.*, **27**, 887-906.

Kiefer, J., and J. Wolfowitz (1959). Optimum designs in regression problems, *Annals of Math. Stat.*, **30**, 271-294.

Kiefer, J., and J. Wolfowitz (1960). The equivalence of two extremum problems, *Canadian J. of Math.*, **12**, 363-366.

Kiselev, O.N., and B.T. Polyak (1991). Ellipsoidal estimation with respect to a generalized criterion, *Automation and Remote Control*, **52**(9), 1281-1292.

Klatte, R., U.W. Kulisch, C. Lawo, M. Rauch and A. Wiethoff (1993). *C-XSC, A C++ Class Library for Extended Scientific Computing*, Springer-Verlag, Heidelberg.

Klatte, R., U.W. Kulisch, M. Neaga, D. Ratz and C. Ullrich (1992). *Pascal-XSC, Language Reference with Examples*, Springer-Verlag, Heidelberg.

Klema, V.C., and A.L. Laub (1980). The singular value decomposition: its computation and some applications, *IEEE Trans. Autom. Control*, **25**(2), 164-176.

König, H., and D. Pallaschke (1981). On Khachiyan's algorithm and minimal ellipsoids, *Numer. Math.*, **36**, 211-223.

Korn, G.A., and T.M. Korn (1968). *Mathematical Handbook for Scientists and Engineers*, second edition, McGraw-Hill, New York.

Krige, D.G. (1951). *A Statistical Approach to Some Mine Valuation and Allied Problems on the Witwatersrand.* Master's Thesis, University of Witwatersrand.

Krolikowski, A., and P. Eykhoff (1985). Input signals design for system identification: a comparative analysis, in *Prep. 7th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, York, 915-920.

Kulcsár, C. (1995). *Planification d'expériences et commande duale*. Thèse de doctorat en sciences, Université Paris-Sud, Orsay.

Kulcsár, C., L. Pronzato and É. Walter (1994). Optimal experimental design and therapeutic drug monitoring, *Int. J. of Bio-Medical Computing*, **36**, 95-101.

Kulcsár, C., L. Pronzato and É. Walter (1995). A dual control policy for linearly parameterized systems, in *Proc. 3rd European Control Conf.*, Rome, **1**, 55-60.

Kulcsár, C., L. Pronzato and É. Walter (1996). Experimental design for the control of linear state-space systems, in *Prep. 13th IFAC World Congress*, San Francisco, C, 175-180.

Kumar, P.R., and P. Varaiya (1986). *Stochastic Systems: Estimation, Identification and Adaptive Control*, Prentice-Hall, Englewood Cliffs.

Kuntzevich, V.M., and M. Lychak (1992). *Guaranteed Estimates, Adaptation and Robustness in Control Systems*, Springer-Verlag, Berlin.

Kurzhanski, A., and I. Vályi (1991). Guaranteed state estimation for dynamic systems: beyond the overviews, in *Prep. 9th IFAC/IFORS Symp. Identification and System Parameter Estimation*, Budapest, 1033-1037.

Kushner, H.J., and J. Yang (1993). Stochastic approximation with averaging: optimal asymptotic rates of convergence for general processes, *SIAM J. Contr. Optim.*, **31**, 1045-1062.

Kushner, H.J., and J. Yang (1995). Stochastic approximation with averaging and feedback: rapidly convergent "on line" algorithms, *IEEE Trans. Autom. Control*, **40**(1), 24-34.

Lahanier, H., É. Walter and R. Gomeni (1987). OMNE: a new robust membership-set estimator for the parameters of nonlinear models, *J. Pharm. Biopharm.*, **15**, 203-219.

Laird, N. (1978). Nonparametric maximum likelihood estimation of a mixing distribution, *J. Am. Stat. Assoc.*, **73**(364), 805-811.

Landau, I.D. (1979). *Adaptive Control: the Model Reference Approach*. Marcel Dekker, New York.

Landaw, E.M. (1980). *Optimal Experimental Design for Biologic Compartmental Systems with Application to Pharmacokinetics*, Ph.D. dissertation, UCLA, Los Angeles.

Landaw, E.M. (1984). Robust sampling designs for compartmental models under large prior eigenvalue uncertainties, in *Mathematics and Computers in Biomedical Applications* (J. Eisenfeld and C. De Lisi, eds.), Elsevier North-Holland, Amsterdam, 181-187.

Launer R.L., and G.N. Wilkinson (1979). *Robustness in Statistics*, Academic Press, New York.

Lawson, C.L., and R.J. Hanson (1974). *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs.

Lawton, W.H., and E.A. Sylvestre (1971). Elimination of linear parameters in nonlinear regression, *Technometrics*, **13**, 461-467.

Lecourtier, Y, and A. Raksanyi (1987). The testing of structural properties through symbolic computation, in *Identifiability of Parametric Models* (É. Walter, ed.), Pergamon, Oxford, 75-84.

Lemaréchal, C. (1989). *Méthodes numériques d'optimisation*. Fascicule INRIA, Collection Didactique, **6**.

Leontaridis, I.J., and S.A. Billings (1985a). Input-output parametric models for nonlinear systems, Part I: Deterministic non-linear systems. *Int. J. Control*, **41**(2), 303-328

Leontaridis, I.J., and S.A. Billings (1985b). Input-output parametric models for nonlinear systems, Part II: Stochastic non-linear systems. *Int. J. Control*, **41**(2), 329-344.

Leroux, B.G. (1992). Consistent estimation of a mixing distribution, *Annals of Stat.*, **20**(3), 1350-1360.

Levenberg, K. (1944). A method for the solution of certain nonlinear problems in least squares, *Quart. Appl. Math.*, **2**, 164-168.

Levin, A. (1965). On an algorithm for the minimization of convex functions, *Soviet Mathematics Doklady*, **6**(1), 286-290.

Lindley, D.V. (1968). The choice of variables in multiple regression, *J. Royal Statist. Soc.*, **B32**, 31-53.

Lindsay, B.G. (1983). The geometry of mixture likelihoods: a general theory, *Annals of Stat.*, **11**(1), 86-94.

Liu, M.S., M. Nayeri and J.R. Deller, Jr. (1994). Do interpretable optimal bounding ellipsoid algorithms converge? Part 2 — OBE vs. RLS: clearing the smoke, in *Prep. 10th IFAC Symp. on System Identification* (M. Blanke and T. Söderström, eds.), Copenhagen, **3**, 395-400.

Ljung, L. (1979). Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems, *IEEE Trans. Autom. Control*, **24**, 36-50.

Ljung, L. (1987). *System Identification, Theory for the User*, Prentice-Hall, Englewood Cliffs.

Ljung, L., and T. Glad (1994). On global identifiability for arbitrary model parametrizations, *Automatica*, **30**, 265-276.

Ljung , L. and H. Hjalmarsson (1995). System identification through the eyes of model validation, in *Proc. 3rd European Control Conf.*, Rome, 949-954.

Ljung, L., and T. Söderström (1983). *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA.

Ljung, L., T. Söderström and I. Gustavsson (1975). Counterexamples to general convergence of a commonly used recursive identification method, *IEEE Trans. Autom. Control*, **20**(5), 643-652.

Loeb, J. (1967), *Identification expérimentale des processus industriels*, Dunod, Paris.

Luenberger, D.G. (1973). *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA.

Macchi, O. (1995). *Adaptive Signal Processing: the Least Mean Squares Approach, with Applications in Transmission*, Wiley, New York.

Madansky, A. (1988). *Prescriptions for Working Statisticians*, Springer-Verlag, Berlin.

Maksarov, D.G., and J.P. Norton (1996a). Tuning of noise bounds in state bounding, in *Proc. CESA 1996 IMACS IEEE-SMC Multiconference*, Lille, Symposium on Modelling, Analysis and Simulation, **2**, 837-842.

Maksarov, D.G., and J.P. Norton (1996b). State bounding with ellipsoidal set description of the uncertainty, *Int. J. Control*, to appear.

Mallet, A. (1983). *Méthodes d'estimation de lois à partir d'observations indirectes d'un échantillon: application aux caractéristiques de population de modèles biologiques*, Thèse de Doctorat d'État, Université de Paris 6.

Mallet, A. (1986). A maximum likelihood estimation method for random coefficient regression models, *Biometrika*, **73**(3), 645-656.

Mallet, A., and F. Mentré (1988). An approach to the design of experiments for estimating the distribution of parameters in random models, in *Prep. 12th IMACS World Congress on Scientific Computation*, Paris, **5**, 134-137.

Mandal, N.K. (1989). D-optimal designs for estimating the optimum point in a quadratic response surface — rectangular region, *J. Statist. Planning and Inference*, **23**, 243-252.

Markus, M.B., and J. Sacks (1977). Robust designs for regression problems, in *Statistical Decision Theory and Related Topics II*, Academic Press, New York, 245-268.

Marquardt, D.W. (1963). An algorithm for least-squares estimation of non-linear parameters, *J. SIAM*, **11**, 431-441.

Marquardt, D.W. (1970). Generalized inverses, ridge regression, biased linear estimation and nonlinear estimation, *Technometrics*, **12**, 591-612.

Matheron, G. (1963), Principles of geostatistics, *Economic Geology*, **58**, 1246-1266.

Mathew, T., and K. Nordström (1993). Least squares and least absolute deviation procedures for approximately linear models, *Statistics and Probability Letters*, **16**, 153-158..

Mead, R., and D.J. Pike (1975). A review of responsė surface methodology from a biometric viewpoint, *Biometrics*, **31**, 803-851.

Mehra, R.K. (1974a). Optimal inputs for linear system identification, *IEEE Trans. Autom. Control*, **19**, 192-200.

Mehra, R.K. (1974b). Optimal input signals for parameter estimation in dynamic systems, survey and new results, *IEEE Trans. Autom. Control*, **19**, 753-768.

Mehra, R.K. (1981). Choice of input signals, in *Trends and Progress in System Identification* (P. Eykhoff, ed.), Pergamon, Oxford, 305-366.

Melas, V.B. (1978). Optimal designs for exponential regressions, *Math. Operationsforsch. und Statist., Ser. Statistics*, **9**, 753-768.

Mentré, F. (1984). *Apprentissage de la loi de probabilité des paramètres d'un modèle par approximation stochastique*, Thèse de Doctorat de 3ème cycle, Université de Paris 7.

Mentré, F., P. Burtin, Y. Merlé, J. Van Bree, A. Mallet and J.-L. Steimer (1995). Sparse sampling optimal designs in pharmacokinetics and toxicokinetics, *Drug Information J.*, **29**, 997-1019.

Mentré, F., A. Mallet and J.-L. Steimer (1988). Hyperparameter estimation using stochastic approximation with application to population pharmacokinetics, *Biometrics*, **44**, 673-683.

Merkuryev, Y.A. (1989). Identification of objects with unknown bounded disturbances, *Int. J. Control*, **50**, 2333-2340.

Messaoud, H., G. Favier and R. Santos Mendes (1992). Adaptive robust pole placement by connecting identification and control, in *Prep. 4th IFAC Int. Symp. Adaptive Systems in Control and Signal Processing*, Grenoble, 41-46.

Middleton, R.H., and G.C. Goodwin (1990). *Digital Control and Estimation, a Unified Approach*, Prentice-Hall, Englewood Cliffs.

Milanese, M., and G. Belforte (1982). Estimation theory and uncertainty intervals evaluation in presence of unknown but bounded errors: linear families of models and estimators, *IEEE Trans. Autom. Control*, **27**, 408-414.

Milanese, M., J.P. Norton, H. Piet-Lahanier and É. Walter (eds.) (1996). *Bounding Approaches to System Identification*, Plenum Press, New York.

Minoux, M. (1983). *Programmation mathématique, théorie et algorithmes*, **1**, Dunod, Paris.

Mitchell, T.J. (1974). An algorithm for the construction of "D-optimal" experimental designs, *Technometrics*, **16**, 203-210.

Mo, S.H., and J.P. Norton (1990). Fast and robust algorithms to compute exact polytope parameter bounds, *Math. and Comput. in Simulation*, **32**, 481-493.

Mockus, J. (1989). *Bayesian Approach to Global Optimization, Theory and Applications*, Kluwer, Dordrecht.

Mohammad-Djafari, A., and G. Demoment (1988). Utilisation de l'entropie dans les problèmes de restauration et de reconstruction d'images, *Traitement du Signal*, **5**(4), 235-248.

Mohammad-Djafari, A., and G. Demoment, eds., (1993). *Maximum Entropy and Bayesian Methods*, Kluwer, Dordrecht.

Montgomery, D.C. (1976). *Design and Analysis of Experiments*, Wiley, New York.

Moore, R.E. (1979). *Methods and Applications of Interval Analysis*, SIAM, Philadelphia.

Moore, R.E. (1992). Parameter sets for bounded-error data, *Math. and Comput. in Simulation*, **34**(2), 113-119.

Müller, W.G., and B.M. Pötscher (1992). Batch sequential design for a nonlinear estimation problem, in *Model-Oriented Data Analysis, a Survey of Recent Methods* (V.V. Fedorov, W.G. Müller and I.N. Vuchkov, eds.), Physica-Verlag, Heidelberg, 77-87.

Myers, R.H. (1976). *Response Surface Methodology*, Allyn and Bacon, Boston.

Nayeri, M., M.S. Liu and J.R. Deller, Jr. (1994). Do interpretable optimal bounding ellipsoid algorithms converge? Part 1 — The long-awaited set-convergence proof, in *Prep. 10th IFAC Symp. on System Identification* (M. Blanke and T. Söderström, eds.), **3**, 389-394.

Nelder, J.A., and R. Mead (1965). A simplex method for function minimization, *Computer J.*, **7**, 308-313.

Nesterov, Y., and A. Nemirovskii (1994). *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia.

Neuman, C.P., and A.K. Sood (1971). Sensitivity functions for multi-input, linear time-invariant systems, *Int. J. Control*, **13**, 1137-1150.

Neuman, C.P., and A.K. Sood (1972). Parameter sensitivity in linear systems, a controllability viewpoint, *Proc. IEE*, **119**, 1217-1219.

Norton, J.P. (1986a). *An Introduction to Identification*, Academic Press, London.

Norton, J.P. (1986b). Problems in identifying the dynamics of biological systems from very short records, in *Proc. 25th IEEE Conf. on Decision and Control*, Athens, 286-290.

Norton, J.P. (1987). Identification of parameter bounds for ARMAX models from records with bounded noise, *Int. J. Control*, **45**, 375-390.

Norton, J.P. (1989). Recursive computation of inner bounds for the parameters of linear models, *Int. J. Control*, **50**, 2423-2430.

Norton, J.P. (ed.) (1994). Special issue on bounded-error estimation, **1**, *Int. J. Adapt. Control Signal Process.*, **8**(1), 1-118.

Norton, J.P. (ed.) (1995). Special issue on bounded-error estimation, **2**, *Int. J. Adapt. Control Signal Process.*, **9**(1), 1-132.

Norton, J.P., and S.H. Mo (1990). Parameter bounding for time varying systems, *Math. and Comput. in Simulation*, **32**, 527-534.

Norton, J.P., and S.M. Veres (1991). Identification of nonlinear state-space models by deterministic search, in *Prep. 9th IFAC/IFORS Symp. on Identification and System Parameter Estimation* (Cs. Bányász and L. Keviczky, eds.), Budapest, **1**, 363-368.

Obali, H. (1993). Personal communication.

O'Hagan, A. (1978). Curve fitting and optimal design for prediction (with discussion), *J. Royal Statist. Soc.*, **B40**(1), 1-42.

Ollivier, F. (1990). *Le problème de l'identifiabilité structurelle globale: approche théorique, méthodes effectives et bornes de complexité.* Thèse de Doctorat en Sciences, École Polytechnique, Palaiseau.

Orfanidis, S.J. (1988). *Optimum Signal Processing, An Introduction*, second edition, McGraw-Hill, New York.

Osborne, M.R., and G.A. Watson (1971). On an algorithm for discrete nonlinear L_1 approximation, *Comput. J.*, **14**, 184-188.

Panuska, V. (1968). A stochastic approximation method for identification of linear systems using adaptive filtering, in *Proc. Joint Autom. Control Conf.*, Ann Arbor, 1014-1021.

Pázman, A. (1984). Probability distribution of the multivariate nonlinear least squares estimates, *Kybernetika* (Prague), **20**, 209-230.

Pázman, A. (1986). *Foundations of Optimal Experimental Design*, VEDA, Bratislava; Reidel, Dordrecht.

Pázman, A. (1990). Small-sample distributional properties of nonlinear regression estimators (a geometric approach) (with discussion), *Statistics*, **21**(3), 323-367.

Pázman, A. (1993). *Nonlinear Statistical Models*, Kluwer, Dordrecht.

Pázman, A., and L. Pronzato (1992a). Nonlinear experimental design based on the distribution of estimators, *J. of Statistical Planning and Inference*, **33**, 385-402.

Pázman, A., and L. Pronzato (1992b). Nonlinear experimental design for constrained LS estimation, in *Model-Oriented Data Analysis, a Survey of Recent Methods* (V.V. Fedorov, W.G. Müller and I.N. Vuchkov, eds.), Physica-Verlag, Heidelberg, 67-76.

Pázman, A., and L. Pronzato (1994). Densities of nonlinear functions of the nonlinear least-squares estimator, in *Prep. 10th IFAC Symp. on System Identification* (M. Blanke and T. Söderström, eds.), Copenhagen, **1**, 163-168.

Pázman, A., and L. Pronzato (1996). A Dirac-function method for densities of nonlinear statistics and for marginal densities in nonlinear regression, *Stat. and Prob. Letters*, **26**, 159-167.

Pearce, S.C. (1983). *The Agricultural Field Experiment: A Statistical Examination of Theory and Practice*, Wiley, Chichester.

Penenko, V.V. (ed.) (1981). *Mathematical Methods for Experiment Design*, Nauka, Novosibirsk (in Russian).

Pesotchinsky, L. (1982). Optimal robust designs: linear regression in $\mathbb{R}^k$, *Annals of Stat.*, **10**(2), 511-525.

Picard, D., and B. Prum (1992). The bias of the MLE, an example of behaviour of different corrections in a genetic model, *Statistics*, **23**, 159-169.

Pichat, M., and J. Vignes (1993). *Ingénierie du contrôle de la précision des calculs sur ordinateurs*, Technip, Paris.

Piet-Lahanier, H., and É. Walter (1990). Characterization of non-connected parameter uncertainty regions, *Math. and Comput. in Simulation*, **32**, 553-560.

Piet-Lahanier, H., and É. Walter (1993). Polyhedric approximation and tracking for bounded-error models, in *Proc. IEEE Int. Symp. Circuits and Systems*, Chicago, 782-785.

Piet-Lahanier, H., and É. Walter (1994). Bounded-error tracking of time-varying parameters, *IEEE Trans. on Autom. Control*, **39**, 1661-1664.

Pilz, J. (1983). *Bayesian Estimation and Experimental Design in Linear Regression Models*, Teubner-Texte zur Mathematik, **55**, Leipzig ; Wiley, New York, 1991.

Pohjanpalo, H. (1978). System identifiability based on the power series expansion of the solution, *Math. Biosci.*, **41**, 21-33.

Polak, E. (1971). *Computational Methods in Optimization*, Academic Press, New York.

Polyak, B.T. (1966). A general method for solving extremum problems, *Soviet Mathematics*, **8**, 593-597.

Polyak, B.T. (1987). *Introduction to Optimization*, Optimization Software, New York.

Polyak, B.T., and A.B. Juditsky (1992). Acceleration of stochastic approximation by averaging, *SIAM J. Contr. Optim.*, **30**, 838-855.

Polyak, B.T., and Ya.Z. Tsypkin (1973). Pseudogradient adaptation and training algorithms, *Automation and Remote Control*, **34**, 377-397.

Polyak, B.T., and Ya.Z. Tsypkin (1980). Robust identification, *Automatica*, **16**, 53-63.

Ponce de Leon, A.C.M., and A.C. Atkinson (1991a). Optimum experimental design for discriminating between two rival models in the presence of prior information, *Biometrika*, **78**(3), 601-608.

Ponce de Leon, A.C.M., and A.C. Atkinson (1991b). Optimal design for discriminating between two rival binary data models in the presence of prior information, in *Proc. Probastat'91*, Bratislava, (A. Pázman and J. Volaufová, eds.), Slovak Ak. Sciences, 123-129.

Powell, M.J.D. (1976). Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in *Nonlinear Programming, SIAM, AMS Proc.*, **IX**, (R.W. Cottle and C.E. Lemke, eds.), New York, 53-72.

Powell, M.J.D. (1981). *Nonlinear Optimization*, Academic Press, New York.

Press, W.H., B.P. Flannery, S.A. Teukolsky and W.T. Vetterling (1986). *Numerical Recipes, the Art of Scientific Computing*, Cambridge University Press, Cambridge.

Pronzato, L., C.Y. Huang, É. Walter, Y. Le Roux and A. Frydman (1989). Planification d'expériences pour l'estimation de paramètres pharmacocinétiques, *Actes du Colloque Informatique et Médicaments*, Hôpital Cochin, Paris, Springer-Verlag, Paris, 3-13.

Pronzato, L. , C. Kulcsár and É. Walter (1996). An actively adaptive control policy for linear models, *IEEE Trans. on Autom. Control*, **41**(6), 855-858.

Pronzato, L., and A. Pázman (1994a). Bias correction in nonlinear regression via two-stage least-squares estimation, in *Prep. 10th IFAC/IFORS Symp. on System Identification* (M. Blanke and T. Söderström, eds.), Copenhagen, **1**, 137-142.

Pronzato, L. and A. Pázman (1994b). Second-order approximation of the entropy in nonlinear least-squares estimation, *Kybernetika*, **30**(2), 187-198 (erratum in **32**(1), 104, 1996).

Pronzato, L., and É. Walter (1985). Robust experiment design via stochastic approximation, *Math. Biosci.*, **75**, 103-120.

Pronzato, L., and É. Walter (1988). Robust experiment design via maximin optimization, *Math. Biosci.*, **89**, 161-176.

Pronzato, L., and É. Walter (1990). Experiment design for bounded-error models, *Math. and Comput. in Simulation*, **32**, 571-584.

Pronzato, L., and É. Walter (1991a). Robustness to outliers of bounded-error estimators, consequences on experiment design, in *Prep. 9th IFAC/IFORS Symp. Identification and System Parameter Estimation*, Budapest, 821-826.

Pronzato, L., and É. Walter (1991b). Sequential experimental design for parameter bounding, in *Proc. 1st European Control Conf.*, Grenoble, 1181-1186.

Pronzato, L., and É. Walter (1991c). Détermination bayesienne des conditions opératoires maximisant un critère de qualité, *Statistique et Analyse des Données*, **16**(2), 107-125.

Pronzato, L., and É. Walter (1992a). Experimental design for estimating the optimum point in a response surface, *Stochastic Optimization and Design*, 1(1), 21-49. Reprinted in 1993 in *Acta Applicandae Mathematicae*, **33**, 45-68.

Pronzato, L., and É. Walter (1992b). Nonsequential Bayesian experimental design for response optimization, in *Model-Oriented Data Analysis, a Survey of Recent Methods* (V.V. Fedorov, W.G. Müller and I.N. Vuchkov, eds.), Physica-Verlag, Heidelberg, 89-102.

Pronzato, L., and É. Walter (1993). Volume-optimal inner and outer ellipsoids, with application to parameter bounding, in *Proc. 2nd European Control Conf.*, Groningen, 258-263.

Pronzato, L., and É. Walter (1994a). Minimal volume ellipsoids, *Int. J. Adapt. Control Signal Process.*, **8**(1), 15-30.

Pronzato, L., and É. Walter (1994b). Minimum-volume ellipsoids containing compact sets: application to parameter bounding, *Automatica*, **30**(11), 1731-1739.

Pronzato, L., and É. Walter (1996a). Volume-optimal inner and outer ellipsoids, in *Bounding Approaches to System Identification* (M. Milanese *et al.*, eds.), Plenum Press, New York, 119-138.

Pronzato, L., and É. Walter (1996b). Comments on "An optimal volume ellipsoid algorithm for parameter set estimation", *Internal Report* 96-15, Laboratoire I3S, 250 rue A. Einstein, Sophia Antipolis, 06560 Valbonne, France.

Pronzato, L., and É. Walter (1996c). Robustness to outliers of bounded-error estimators and consequences on experiment design, in *Bounding Approaches to System Identification* (M. Milanese *et al.*, eds.), Plenum Press, New York, 199-212.

Pronzato, L., É. Walter and C. Kulcsár (1993). A dynamical-system approach to sequential design, in *Model-Oriented Data Analysis* (W.G. Müller, H.P. Wynn and A.A. Zhigljavsky, eds.), Physica-Verlag, Heidelberg, 11-24.

Pronzato, L., É. Walter and H. Piet-Lahanier (1989). Mathematical equivalence of two ellipsoidal algorithms for bounded-error estimation, in *Proc. 28th IEEE Conf. on Decision and Control*, Tampa, 1952-1955.

Pronzato, L., É. Walter, A. Venot and J.-F. Lebruchec (1984). A general purpose global optimizer: implementation and applications, *Math. and Comput. in Simulation*, **26**, 412-422.

Pronzato, L., H.P. Wynn and A.A. Zhigljavsky (1995). Dynamic systems in search and optimization, *CWI Quarterly*, **8**(3), 201-236.

Pronzato, L., and A.A. Zhigjavsky (1993), On average-optimal quasi-symmetrical univariate optimization algorithms, in *Model-Oriented Data Analysis* (W.G. Müller, H.P. Wynn and A.A. Zhigljavsky, eds.), Physica-Verlag, Heidelberg, 269-278.

Pshenichnyy, B.N., and V.G. Pokotilo (1983), A minimax approach to estimation of linear regression parameters, *Engrg. Cybernetics*, 77-85.

Pukelsheim, F. (1993). *Optimal Experimental Design*, Wiley, New York.

Pukelsheim, F., and S. Reider (1992). Efficient rounding of approximate designs, *Biometrika*, **79**(4), 763-770.

Quenouille, M. (1949). Approximate tests of correlation in time series, *J. Royal Statist. Soc.*, **B11**, 18-84.

Rabitz, H., M. Kramer and D. Dacol (1983). Sensitivity analysis in chemical kinetics, *Ann. Rev. Phys. Chem.*, **34**, 419-461.

Raksanyi, A., Y. Lecourtier, É. Walter and A. Venot (1985). Identifiability and distinguishability testing via computer algebra, *Math. Biosci.*, **77**(1-2), 245-266.

Rasch, D. (1988). Recent results in growth analysis, *Statistics*, **19**(4), 585-604.

Ratkowsky, D.A. (1983). *Nonlinear Regression Modelling*, Marcel Dekker, New York.

Ratschek, H., and J. Rokne (1988). *New Computer Methods for Global Optimization*, Ellis Horwood, Chichester (distributed by Wiley, New York).

Redner, R.A., and H.E. Walker (1984). Mixture densities, maximum likelihood and the EM algorithm, *SIAM Review*, **26**(2), 195-239.

Reilly, P.M. (1970). Statistical methods in model discrimination, *Canadian J. Chem. Eng.*, **48**, 168-173.

Rényi, A. (1966). *Calcul des probabilités*, Dunod, Paris.

Richalet, J. (1991). *Pratique de l'identification*, Hermès, Paris.

Richalet, J., A. Rault and R. Pouliquen (1971). *Identification des processus par la méthode du modèle*, Gordon and Breach, Paris.

Ritt, J. F. (1950). *Differential Algebra*, American Mathematical Society, Providence.

Robbins, H., and S. Monro (1951). A stochastic approximation method, *Annals of Math. Stat.*, **22**, 400-407.

Rosen, J.B. (1960). The gradient projection method for nonlinear programming, part 1: linear constraints, *SIAM J.*, **8**, 181-217.

Rosen, J.B. (1961). The gradient projection method for nonlinear programming, part 2: nonlinear constraints, *SIAM J.*, **9**, 514-532.

Rousseeuw, P.J., and A.M. Leroy (1987). *Robust Regression and Outlier Detection*, Wiley, New York.

Rubel, L.A., A universal differential equation (1981). *Bull. Am. Math. Soc.*, **4**(3), 345-349.

Sacks, J., and S.B. Schiller (1988). Spacial designs, in *Statistical Decision Theory and Related Topics IV* (S.S. Gupta and J.O. Berger, eds.), Springer-Verlag, Berlin, **2**, 385-399.

Sacks, J., S.B. Schiller and W.J. Welch (1989). Design for computer experiments, *Technometrics*, **31**, 41-47.

Sacks, J., W.J. Welch, T.J. Mitchell and H.P. Wynn (1989). Design and analysis of computer experiments (with discussion), *Statistical Science*, **4**(4), 409-435.

Sacks, J., and D. Ylvisaker (1978). Linear estimation for approximately linear models, *Annals of Stat.*, **6**(5), 1122-1137.

Sacks, J., and D. Ylvisaker (1984). Some model robust designs in regression, *Annals of Stat.*, **12**(4), 1324-1348.

Saridis, G.N. (1974). Stochastic approximation methods for identification and control, a survey, *IEEE Trans. Autom. Control*, **19**, 798-809.

Scheffé, H. (1958). Experiments with mixtures, *J. Royal Statist. Soc.*, **B21**, 344-360.

Schulz, M., and L. Endrenyi (1983). Design of experiments for estimating parameters with unknown heterogeneity of the error variance, in *Proc. Am. Stat. Assoc. Comput. Sect.*, 177-181.

Schumitzky, A. (1991). Nonparametric EM algorithms for estimating prior distributions, *Applied Math. and Computation*, **45**, 143-157.

Schwabe, R. (1995). Designing experiments for additive nonlinear models, in *Advances in Model-Oriented Data Analysis, Proc. MODA4* (Ch.P. Kitsos and W.G. Müller, eds.), Physica Verlag, Heidelberg, 77-85.

Schwabe, R. (1996). *Optimum Designs for Multi-Factor Models*, Springer-Verlag, Heidelberg.

Schweppe, F.C. (1968). Recursive state estimation: unknown but bounded errors and systems inputs, *IEEE Trans. Autom. Control*, **13**, 22-29

Schweppe, F.C. (1973). *Uncertain Dynamic Systems*, Prentice-Hall, Englewood Cliffs.

Searle, S.R. (1971). *Linear Models*, Wiley, New York.

Seber, G.A.F., and C.J. Wild (1989). *Nonlinear Regression*, Wiley, New York.

Sheiner, L.B., and S.L. Beal (1980). Evaluation of methods for estimating population pharmacokinetic parameters: I. Michaelis-Menten model, routine clinical pharmacokinetic data, *J. Pharm. Biopharm.*, **8**(6), 553-571.

Shewry, M.C., and H.P. Wynn (1987). Maximum entropy sampling, *J. Applied Statist.*, **14**, 165-170.

Shewry, M.C., and H.P. Wynn (1988). Maximum entropy sampling and simulation codes, in *Prep. 12th IMACS World Congress on Scientific Computation*, Paris, **2**, 517-519.

Shimizu, K., and E. Aiyoshi (1980). Necessary conditions for min-max problems and algorithm by a relaxation procedure, *IEEE Trans. Autom. Control*, **25**, 62-66.

Shor, N.Z. (1985). *Minimisation Methods for Non-Differentiable Functions*, Springer-Verlag, Berlin.

Sibson, R. (1972). Discussion of Dr. Wynn's paper, *J. Royal Statist. Soc.*, **B34**, 174-175.

Silvey, S.D. (1980). *Optimal Design*, Chapman and Hall, London.

Silvey, S.D., D.M. Titterington and B. Torsney (1978). An algorithm for optimal designs on a finite design space, *Communications in Statistics*, **A7**(14), 1379-1389.

Skovgaard, I.M. (1985). Large deviation approximations from maximum likelihood estimators, *Probab. Math. Statist.*, **6**, 89-107.

Snee, R.D. (1977). Validation of regression models. Methods and examples. *Technometrics*,**19**, 415-428.

Söderström, T (1977). On model structure testing in system identification, *Int. J. Control*, **26**, 1-18.

Söderström, T., and P. Stoica (1983). *Instrumental Variable Methods for System Identification*, Springer-Verlag, Berlin.

Sorenson, H.W. (1980). *Parameter Estimation, Principles and Problems*, Marcel Dekker, New York.

Spall, J.C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE Trans. Autom. Control*, **37**(3), 332-341.

Spall, J.C. (1995). Stochastic version of second-order (Newton-Raphson) optimization using only function measurements, in *Proc. Winter Simulation Conf.* (C. Alexopoulas, K. Kang, W.R. Lilegdon and D. Goldsman, eds.), 347-352.

Speelpening, B. (1980). *Compiling Fast Partial Derivatives of Functions Given by Algorithms*, Ph.D. dissertation, Department of Computer Science, University of Illinois at Urbana Champaign.

Steiglitz, K., and L.E. McBride (1965). A technique for the identification of linear systems. *IEEE Trans. on Autom. Control*, **10**, 461-464.

Steimer, J.-L., A. Mallet, J.-L. Golmard and J.-F. Boisvieux (1984). Alternative approaches to estimation of population pharmacokinetic parameters, comparison with NONMEM, *Drug Metab. Review*, **15**, 265-292.

Steinberg, D.M. (1985). Model robust response surface designs: scaling two-level factorials, *Biometrika*, **72**(3), 513-526.

Steinberg, D.M., and W.G. Hunter (1984). Experimental design: review and comment (with discussion), *Technometrics*, **26**(2), 71-130.

Stigler, S.M. (1971). Optimal experimental design for polynomial regression, *J. Am. Statist. Assoc.*, **66**, 311-318.

Stigler, S.M. (1981). Gauss and the invention of least-squares, *Annals of Stat.*, **9**(4), 465-474.

StJohn, R.C., and N.R. Draper (1975). D-optimality for regression designs: a review, *Technometrics*, **17**(1), 15-23.

Stoica, P., and T. Söderström (1981). The Steiglitz-Mcbride identification algorithm revisited, convergence analysis and accuracy aspects, *IEEE Trans. Autom. Control*, **26**, 712-717.

Stone, M. (1974). Cross-validity choice and assessment of statistical predictors. *J. Royal Statist. Soc.* **B36**: 111-147.

Street, A.P. (1987). *Combinatorics of Experimental Design*, Clarendon Press, Oxford.

Strobach, P. (1991). New forms of Levinson and Schur algorithms, *IEEE Signal Processing Magazine*, **1**, 12-36.

Studden, W.J. (1982). Some robust-type D-optimal designs in polynomial regression, *J. Am. Statist. Assoc.*, **77**, 916-921.

Sussman, H. J. (1977). Existence and uniqueness of minimal realizations of nonlinear systems. *Math. Syst. Theory*, **10**, 263-284.

Talagrand, O. (1991). The use of adjoint equations in numerical modelling of the atmospheric circulation, *Automatic Differentiation of Algorithms : Theory, Implementation and Application.* (A. Griewank and G. Corliss, eds.), SIAM, Philadelphia, 169-180.

Talmon, J.L., and A.J.W. van den Boom (1973). On the estimation of the transfer function parameters of process and noise dynamics using a single stage estimator, in *Proc. 3rd IFAC Symp. Identification and System Parameter Estimation*, The Hague, North-Holland, Amsterdam, 711-720.

Tarasov, S.P., L.G. Khachiyan and I.I. Èrlikh (1988). The method of inscribed ellipsoids, *Soviet Math. Dokl.*, **37**(1), 226-230.

Thomson, A.H., and B. Whiting (1992). Bayesian parameter estimation and population pharmacokinetics, *Clin. Pharmacokinet.*, **22**(6), 447-467.

Thompson, A.M., J.C. Brown, J.W. Kay and D.M. Titterington (1991). A study of methods of choosing the smoothing parameter in image restoration by regularization, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **13**(4), 326-339.

Titterington, D.M. (1976). Algorithms for computing D-optimal designs on a finite design space, in *Proc. Conf. on Information Science and Systems*, John Hopkins Univ., Baltimore, 213-216.

Titterington, D.M. (1978). Estimation of correlation coefficients by ellipsoidal trimming. *J. of Royal Statist. Soc.*, **C27**(3), 227-234.

Titterington, D.M. (1980). Aspects of optimal design in dynamic systems, *Technometrics*, **22**(3), 287-299.

Topkis, D.M., and A.F. Veinnot (1967). On the convergence of some feasible direction algorithms for nonlinear programming, *SIAM J. Control*, **5**, 268-279.

Torsney, B. (1983). A moment inequality and monotonicity of an algorithm, in *Proc. Int. Symp. on Semi-Infinite Programming and Applications*, Austin (K.O. Kortanek and A.V. Fiacco, eds.), Springer-Verlag, Heidelberg, 249-260.

Torsney, B. (1988). Computing optimizing distributions with applications in design, estimation and image processing, in *Optimal Design and Analysis of Experiments* (Y. Dodge, V.V. Fedorov and H.P. Wynn, eds.), North-Holland, Amsterdam, 361-370.

Torsney, B. and A.M. Alahmadi (1992). Further developments of algorithms for constructing optimizing distributions, in *Model-Oriented Data Analysis, a Survey of Recent Methods* (V.V. Fedorov, W.G. Müller and I.N. Vuchkov, eds.), Physica-Verlag, Heidelberg, 121-129.

Tsypkin, Ya.Z. (1983). Optimality in identification of linear plants, *Int. J. Systems Sci.*, **14**(1), 59-74.

Tsypkin, Ya.Z., and V.A. Lototsky (1985). Optimal identification algorithms and their application to process control and inventory systems, in *Prep. 7th IFAC/IFORS Symp. Identification and System Parameter Estimation*, York, 1535-1540.

Unbehauen, H., and G.P. Rao (1987). *Identification of Continuous Systems*, North-Holland, Amsterdam.

Vajda, S., K. R. Godfrey and H. Rabitz (1989). Similarity transformation approach to identifiability analysis of nonlinear compartmental models. *Math. Biosci.*, **93**, 217-248.

Vajda, S., and H. Rabitz (1989) State isomorphism approach to global identifiability of nonlinear systems. *IEEE Trans. Autom. Control*, **AC-34**, 220-223.

Vajda, S., H. Rabitz, É. Walter and Y. Lecourtier (1989). Qualitative and quantitative identifiability analysis of nonlinear chemical models. *Chem. Eng. Comm.*, **83**, 191-219.

Valko, P., and S. Vajda (1984). An extended ODE solver for sensitivity calculations, *Computers and Chemistry*, **8**(4), 255-271.

Vandanjon, P.-O. (1995). *Identification robuste des paramètres inertiels des bras manipulateurs par découplage fréquentiel*, Thèse de Doctorat en Sciences, École des Mines de Paris.

Van Huffel, S. (1987). *Analysis of the Total Least Squares Problem and its Use in Parameter Estimation*, Ph.D. dissertation, Katholieke Universiteit Leuven.

Van Overschee, P., and B. De Moor (1994). N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, **30**, 75-93.

Van Overschee, P., and B. De Moor (1996). *Subspace Identification for Linear Systems*, Kluwer, Dordrecht.

Velev, K. (1988). Model-reference control of systems with signal dependent parameters using orthonormal functions, in *Proc. 33 Intern. Wiss. Koll. T.H. Ilmenau, Vortragsreihe, Technische Kybernetik/Automatisierrungstechnik*, 309-312.

Venot, A., L. Pronzato, É. Walter and J.-F. Lebruchec (1986). A distribution-free criterion for robust identification with applications in system modelling and image processing, *Automatica*, **22**, 105-109.

Veres, S.M. (1991). *Structure Selection of Stochastic Dynamic Systems: the Information Criterion Approach*, Gordon and Breach, New York.

Veres, S.M., and J.-P. Norton (1991a). Structure selection for bounded-parameter models: consistency conditions and selection criterion, **36**(4), 474-481.

Veres, S.M., and J.-P. Norton (1991b). Adaptive pole-placement control using parameter bounds, in *Proc. 30th IEEE Conf. Decision and Control*, Brighton, 2860-2865.

Veres, S.M., and J.-P. Norton (1991c). Parameter bounding algorithms for linear errors in variables models, in *Prep. 9th IFAC/IFORS Symp. Identification and System Parameter Estimation*, Budapest, 1038-1043.

Vetter , W.J. (1970). Derivative operations on matrices, *IEEE Trans. Autom. Control*, **15**, 241-244.

Vetter, W.J. (1973). Matrix calculus operations and Taylor expansions, *SIAM Review*, **15**(2), 352-369.

Vicino, A., and G. Zappa (1992). Sequential approximation of uncertainty sets via parallelotopes, in *Proc. Workshop on the Modeling of Uncertainty in Control Systems*, Springer-Verlag, Heidelberg.

Vignes, J. (1969). *Étude et mise en œuvre d'algorithmes de recherche d'un extremum d'une fonction de plusieurs variables*, Thèse de Doctorat d'État, Université de Paris.

Vignes, J. (1978). New methods for evaluating the validity of the results of mathematical computations, *Math. and Comput. in Simulation*, **20**, 227-249.

Vignes, J. (1988). Review on stochastic approach to round-off error analysis and its applications, *Math. and Comput. in Simulation*, **30**, 481-491.

Vignes, J., with R. Alt and M. Pichat (1980). *Algorithmes numériques, analyse et mise en œuvre. Tome 2 : Équations et systèmes non linéaires*. Technip, Paris.

Vila, J.P. (1986). Optimal designs for parameter estimation in nonlinear regression models: exact criteria, in *Proc. 13th Int. Biometric Conf.*, Seattle, Session 13, Paper 3.

Vila, J.P. (1988). New algorithmic and software tools for D-optimal design computation in nonlinear regression, *Compstat 1988*, Physica-Verlag, Heidelberg, 409-414.

Vila, J.P. (1990). Exact experimental designs via stochastic optimization for nonlinear regression models, *Compstat 1990*, Physica-Verlag, Heidelberg, 291-296.

Wahlberg, B. (1991). System identification using Laguerre models, *IEEE Trans. Autom. Control*, **36**(5), 551-562.

Wahlberg, B. (1994). System identification using Kautz models, *IEEE Trans. Autom. Control*, **39**, 1276-1282.

Walter, É. (1982). *Identifiability of State-Space Models*, Lecture Notes in Biomathematics, Springer-Verlag, Berlin.

Walter, É., (ed.) (1987). *Identifiability of Parametric Models*, Pergamon Press, Oxford.

Walter, É. (ed.) (1990). Special issue on Parameter Identification with Error Bounds, *Math. and Comput. in Simulation*, **32**(5&6), 447-607.

Walter, É., and Y. Lecourtier (1981). Unidentifiable compartmental models: what to do? *Math. Biosci.*, **56**, 1-25.

Walter, É., Y. Lecourtier and J. Happel (1984). On the structural output distinguishability of parametric models, and its relation with structural identifiability, *IEEE Trans. Autom. Control*, **29**, 56-57.

Walter, É., Y. Lecourtier, J. Happel and J.-Y. Kao (1986). Identifiability and distinguishability of fundamental parameters in catalytic methanation, *AIChE J.*, **32**(8), 1360-1366.

Walter, É., Y. Lecourtier, A. Raksanyi and J. Happel (1985). On the distinguishability of parametric models with different structures, in *Mathematics and Computers in Biomedical Applications* (J. Eisenfeld and C. De Lisi, eds), North-Holland, Amsterdam, 145-160.

Walter, É., and H. Piet-Lahanier (1988). Estimation of the parameter uncertainty resulting from bounded-error data, *Math. Biosci.*, **92**, 55-74.

Walter, É., and H. Piet-Lahanier (1989). Exact recursive polyhedral description of the feasible parameter set for bounded-error models, *IEEE Trans. Autom. Control*, **34**(8), 911-915.

Walter, É., and H. Piet-Lahanier (1991). Recursive robust minimax estimation for models linear in their parameters, in *Prep. 9th IFAC/IFORS Symp. Identification and System Parameter Estimation*, Budapest, 763-768.

Walter, É., H. Piet-Lahanier and J. Happel (1986). Estimation of nonuniquely identifiable parameters via exhaustive modeling and membership set theory, *Math. and Comput. in Simulation*, **28**, 479-490.

Walter, É., and L. Pronzato (1987). Robust experiment design: between qualitative and quantitative identifiabilities, in *Identifiability of Parametric Models* (É. Walter, ed.), Pergamon, Oxford, 104-113.

Walter, É., and L. Pronzato (1990). Qualitative and quantitative experiment design for phenomenological models: a survey, *Automatica*, **26**, 195-213.

Walter, É., and L. Pronzato (1994). Characterizing sets defined by inequalities, in *Prep. 10th IFAC/IFORS Symp. on System Identification* (M. Blanke and T. Söderström, eds.), Copenhagen, **2**, 15-26.

Walter, É., and L. Pronzato (1995). Identifiabilities and nonlinearities, in *Nonlinear Systems* (A.J. Fossard and D. Normand-Cyrot, eds.), Chapman and Hall, London, **1**, Chapter 3, 111-143.

Walter, É., L. Pronzato and A. Venot (1989). Theoretical properties of sign change criteria for robust off-line estimation, *Automatica*, **25**(6), 949-952.

Welch, W.J. (1983). A mean squared error criterion for the design of experiments, *Biometrika*, **70**(1), 205-213.

Welch, W.J., R.J. Buck, J. Sacks, H.P. Wynn, T.J. Mitchell and M.D. Morris (1992). Screening, predicting and computer experiments, *Technometrics*, **34**(1), 15-25.

Widrow, B., and S.D. Stearns (1985). *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs.

Wilkie, D.F., and W.R. Perkins (1969). Essential parameters in sensitivity analysis, *Automatica*, **5**, 191-198.

Willems, J.C. (1986a). From time series to linear systems, Part 1. Finite dimensional linear time invariant systems, *Automatica*, **22**(5), 561-580.

Willems, J.C. (1986b). From time series to linear systems, Part 2. Exact modelling, *Automatica*, **22**(6), 675-694.

Willems, J.C. (1987). From time series to linear systems, Part 3. Approximate modelling, *Automatica*, **23**(1), 87-115.

Wolfe, P. (1969). Convergence conditions for ascent methods, *SIAM Review*, **11**, 226-235.

Wong, W.K. (1992). A unified approach to the construction of minimax designs, *Biometrika*, **79**(3), 611-619.

Wong, W.K. and R.D. Cook (1993). Heteroscedastic G-optimal designs, *J. Royal Statist. Soc.*, **B55**(4), 871-880.

Wonnacott, T.H., and R.J. Wonnacott (1984). *Introductory Statistics*, Wiley, New York.

Wu, C.-F.J. (1978). Some algorithmic aspects of the theory of optimal designs, *Annals of Stat.*, **6**(6), 1286-1301.

Wu, C.-F.J. (1983). On the convergence properties of the EM algorithm, *Annals of Stat.*, **11**(1), 95-103.

Wu, C.-F.J. (1985). Asymptotic inference from sequential design in a nonlinear situation, *Biometrika*, **72**(3), 553-558.

Wynn, H.P. (1972). Results in the construction of D-optimum experimental designs, *J. Royal Statist. Soc.*, **B34**, 133-147.

Wynn, H.P., and A.A. Zhigljavsky (1993). Chaotic behaviour of search algorithms: introduction, in *Model-Oriented Data Analysis* (W.G. Müller, H.P. Wynn and A.A. Zhigljavsky, eds.), Physica-Verlag, Heidelberg, 199-211.

Young, P.C. (1968). The use of linear regression and related procedures for the identification of dynamic processes, in *Proc. 7th IEEE Symp. Adaptive Processes*, San Antonio, 501-505.

Zacks, S. (1977). Problems and approaches in design of experiments for estimation and testing in non-linear models, in *Multivariate Analysis IV* (P.R. Krishnaiah, ed.), North-Holland, Amsterdam, 209-223.

Zarrop, M.B. (1979). *Optimal Experiment Design for Dynamic System Identification*, Springer-Verlag, Heidelberg.

Zarrop, M.B., and G.C. Goodwin (1975). Comments on "Optimal inputs for system identification", *IEEE Trans. Autom. Control*, 299-300.

Zhigljavsky, A.A. (1991). *Theory of Global Random Search*, Kluwer, Dordrecht.

Zoutendijk G. (1960). *Methods of Feasible Directions*, Elsevier, Amsterdam.

Index