J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*

HOMOTOPY ALGORITHM FOR RICCATI EQUATIONS

In modern control an H^2 cost function is used to measure the response of a controlled system to wide band disturbances while an H^∞ cost function is used to measure the response to narrow band disturbances and can also be used to account for unstructured uncertainties in the design model. To take into account controller processor limitations, it is also desired to prespecify the order of the controller in the design process. Hence, an important problem in modern control design is the synthesis of fixed-order controllers that are optimized subject to these H^2 and H^∞ constraints. This synthesis problem requires the solution of a challenging optimization problem, and this article reviews a solution approach based on homotopy methods.

There are many approaches to solving both full- and reduced-order linear state equation, quadratic objective function, Gaussian noise (LQG) controller design problems with an H^∞ constraint on disturbance attenuation. The Riccati equation based approach enforces the H^∞ constraint by replacing the covariance Lyapunov equation by a Riccati equation whose solution gives an upper bound on H^2 performance. Numerical algorithms, based on homotopy theory, solve the necessary conditions for a minimum of the upper bound on H^2 performance subject to the H^∞ constraint given by the Riccati equation. A summary of the properties of Riccati equations and numerical algorithms for solving them is also included. The homotopy algorithms are based on a minimal parameter formulation: Ly, Bryson, and Cannon's 2 × 2 block parametrization. An overparametrization formulation is also proposed. Numerical experiments suggest that the combination of a globally convergent homotopy method and a minimal parameter formulation applied to the upper bound minimization gives excellent results for mixed-norm H^2/H^∞ synthesis. The nonmonotonicity of homotopy zero curves is demonstrated, proving that algorithms more sophisticated than standard continuation are necessary. To achieve high computational performance the homotopy algorithm is also parallelized to run in distributed environments such as a network of Unix workstations or an Intel Paragon parallel computer. Comparing results on the workstations with the results from the Intel Paragon, the study concludes that utilizing Unix workstations can be a very cost-effective approach to shorten computation time. Furthermore, this economical way to achieve high performance computation can easily be realized and incorporated in a practical industrial design environment.

The Riccati equation is central to modern linear-quadratic estimation and control design. Many problems in control analysis and synthesis can be formulated in terms of Riccati equations, with the H^2/H^∞ mixed-norm controller synthesis problem being one of them.

The H^2/H^∞ mixed-norm controller synthesis problem provides the means for simultaneously addressing H^2 and H^∞ performance objectives. In practice, such controllers provide both robust performance (via suboptimal H^2) and robust stability (via H^∞). Hence mixed-norm synthesis provides a technique for trading off performance and robustness, a fundamental objective in control design.

The H^2/H^∞ mixed-norm problem has been addressed in a variety of settings. The Riccati equation based approach was given in (1,2) which utilized an H^2 cost bound as the basis for an auxiliary minimization problem. Necessary conditions for optimality within a full- and reduced-order fixed-structure setting were then used to characterize feedback control gains. These necessary conditions have the form of coupled Riccati equations in both the full- and reduced-order cases. In related work (3,4), the H^2 cost bound in the case of equalized H^2 and

H^∞ performance weights was shown to be equal to an entropy cost functional. The centralized controller was then shown to yield a full-order controller that optimizes the entropy cost.

An additional treatment in (5) using a bounded power characterization of the H^2 norm obtained both necessary and sufficient conditions for optimality. Finally, a convex optimization approach was developed in (6) for the full-order problem.

The purpose of this article is to develop numerical algorithms for solving the Riccati equation based mixed-norm H^2/H^∞ problem addressed in (1,2). Basically, the modified cost function is optimized subject to constraints, including a coupled Riccati equation. The approach here is based upon homotopy methods which have been applied to related fixed-structure problems in (7,8,9,10,11). Using globally convergent homotopy techniques similar to those applied to the combined H^2/H^∞ model reduction problem (8,9,10,11), and using a controller parametrization suggested by Ly, Bryson, and Cannon, results are obtained for the combined H^2/H^∞ full- and reduced-order controller synthesis problems. Another parametrization, the input normal Riccati form, was used in (9) and its details will not be repeated here. However, such controller parametrizations, which use the minimum possible number of parameters, make structural assumptions about the optimal controller which may not be valid in a particular case. Invalidity of these assumptions manifests itself in numerical instability, and failure to converge. An over-parametrization formulation which does not make structural assumptions is also proposed. However, over-parametrization introduces singularity of the homotopy map at the solution and the algorithm may fail for a high dimensional system.

These homotopy methods utilize the solution of a related easily solved problem as the starting point. In the case of full-order H^2/H^∞ control with unequalized weights, the starting point is provided by the standard LQG solution. For the reduced-order problem, the starting point is obtained by constructing a low authority, nearly nonminimal LQG compensator (12).

The theoretical foundation of all probability-one globally convergent homotopy methods is given by the following definition and theorem from differential geometry.

Definition. *Let $U \subset \boldsymbol{R}^m$ and $V \subset \boldsymbol{R}^p$ be open sets, and let ρ $U \times [0, 1) \times V \to \boldsymbol{R}^p$ be a C^2 map. ρ is said to be transversal to zero if the Jacobian matrix $D\rho$ has full rank on $\rho^{-1}(0)$.*

Transversality Theorem. If $\rho(a, \lambda, x)$ is transversal to zero, then for almost all $a \in U$ the map

$$\rho_a(\lambda, x) = \rho(a, \lambda, x)$$

is also transversal to zero, that is, with probability one the Jacobian matrix $D\rho_a(\lambda, x)$ has full rank on $\rho^{-1}{}_a(0)$.

To solve the nonlinear system of equations

$$f(x) = 0$$

where f: $\mathbf{R}^p \to \mathbf{R}^p$ is a C^2 map, the general probability-one homotopy paradigm is to construct a C^2 homotopy map ρ: $U \times [0, 1) \times \mathbf{R}^p \to \mathbf{R}^p$ such that

(1) $\rho(a, \lambda, x)$ is transversal to zero, for each fixed $a \in U$,
(2) $\rho_a(0, x) = \rho(a, 0, x) = 0$ is trivial to solve and has a unique solution x_0,
(3) $\rho_a(1, x) = f(x)$,
(4) the connected component of $\rho^{-1}{}_a(0)$ containing $(0, x_0)$ is bounded.

Then (from the transversality theorem) for almost all $a \in U$ there exists a zero curve γ of ρ_a, along which the Jacobian matrix $D\rho_a$ has rank p, emanating from $(0, x_0)$ and reaching a zero $\bar{x}$ of f at $\lambda = 1$. This zero curve

γ has no bifurcations (i.e., γ is a smooth 1-manifold), and has finite arc length in every compact subset of $(0, 1) \times \mathbf{R}^p$. Furthermore, if $Df(\bar{x})$ is nonsingular, then γ has finite arc length. The complete homotopy paradigm is now apparent: Construct the homotopy map ρ_a and then track its zero curve γ from the known point $(0, x_0)$ to a solution $\bar{x}$ at $\lambda = 1$. ρ_a is called a probability-one homotopy because the conclusions hold almost surely with respect to a, that is, with probability one. Since the vector a, and indirectly the starting point x_0, are essentially arbitrary, an algorithm to follow the zero curve γ emanating from $(0, x_0)$ until a zero $\bar{x}$ of $f(x)$ is reached (at $\lambda = 1$) is legitimately called *globally convergent.*

There is considerable confusion in the control literature over the terms continuation, homotopy, and globally convergent. A careful discussion of the distinct meanings of these terms can be found in (13). Continuation refers to the standard classical technique of solving $\rho(\theta, \bar{\lambda} + \Delta\lambda) = 0$ with fixed $\Delta\lambda > 0$, given a solution $(\bar{\theta}, \bar{\lambda})$: $\rho(\bar{\theta}, \bar{\lambda}) = 0$. It is implicitly assumed that $\theta = \theta(\lambda)$, that is, the zero curve γ of $\rho(\theta, \lambda)$ being tracked in (θ, λ) space is monotone in λ. Other tacit assumptions are that γ does not bifurcate or otherwise contain singularities. The most general *homotopy* methods make no such assumptions, and include mechanisms to deal with bifurcations and turning points. In particular, homotopy methods do not assume that the zero curve γ is monotone in λ, that is, $\theta = \theta(\lambda)$. *Globally convergent* means that the zero curve γ reaches a solution $\bar{\theta}$, $\rho(\bar{\theta}, 1) = 0$, from an arbitrary starting point θ_0, $\rho(\theta_0, 0) = 0$. A continuation or homotopy algorithm is not a priori globally convergent. A particular class of homotopy methods, known as *probability-one homotopy methods,* are provably globally convergent under mild assumptions (14), and their zero curve γ is guaranteed to contain no singularities with probability one. The homotopy algorithms proposed here are examples of probability-one globally convergent homotopy methods; the matrices $A_0, B_0, \ldots$, and the starting point θ_0 defined later play the role of the parameter vector a in the probability-one homotopy theory (13). Computational results for the example in (1) clearly demonstrate the nonmonotonicity in λ and that standard continuation in λ would fail.

The LQG controller synthesis problem with an H^∞ performance bound can be stated as follows: given the nth order stabilizable and detectable plant

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + D_1 w(t) \\ y(t) &= Cx(t) + D_2 w(t) \end{aligned} \tag{1}$$

where $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{l\times n}$, $D_1 \in \mathbf{R}^{n\times p}$, $D_2 \in \mathbf{R}^{l\times p}$, $D_1 D^T{}_2 = 0$, and $w(t)$ is p-dimensional white noise, find a n_cth order dynamic compensator

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t) + B_c y(t) \\ u(t) &= C_c x_c(t) \end{aligned} \tag{2}$$

where $A_c \in \mathbf{R}^n c_{\times n}{}^c$, B_c ${}^{\in\mathbf{R}}{}_n c_{\times l}$, $C_c \in {}^{\mathbf{R}} m_{\times n}{}^c$, and $n_c \leq n$, which satisfies the following criteria:

(1) The closed-loop system of Eqs. (1) and (2) is asymptotically stable, that is

$$\tilde{A} = \begin{pmatrix} A & BC_c \\ B_c C & A_c \end{pmatrix}$$

is asymptotically stable;

(2) The $q_\infty \times p$ closed-loop transfer function

$$H(s) \equiv \tilde{E}_\infty (sI_{\tilde{n}} - \tilde{A})^{-1} \tilde{D}$$

from $w(t)$ to

$$z(t) = E_{1\infty}x(t) + E_{2\infty}u(t) \tag{3}$$

where $\tilde{C}_\infty = (E_{1\infty}\ E_{2\infty}C_c)$ ($E_{1\infty} \in R^{q_\infty \times n}$, $E_{2\infty} \in R^{q_\infty \times m}$, $E^T{}_{1\infty}E_{2\infty} = 0$), $\tilde{n} = n + n_c$, and

$$\tilde{D} = \begin{pmatrix} D_1 \\ B_c D_2 \end{pmatrix}$$

satisfies the constraint

$$\|H(s)\|_\infty \leq \gamma \tag{4}$$

where $\gamma > 0$ is a given constant; and

(3) The performance functional

$$J(A_c, B_c, C_c) \equiv \lim_{t \to \infty} \mathscr{E}\left[x^T(t)R_1x(t) + u^T(t)R_2u(t)\right] \tag{5}$$

is minimized, where $\mathscr{E}$ is the expected value, $R_1 = E^T{}_1E_1 \in \mathbf{R}^{n\times n}$ and $R_2 = E^T{}_2E_2 \in \mathbf{R}^{m\times m}$ ($E_1 \in \mathbf{R}^{q\times n}$, $E_2 \in \mathbf{R}^{q\times m}$, $E^T{}_1E_2 = 0$) are, respectively, symmetric positive semidefinite and symmetric positive definite weighting matrices.

The closed-loop system of Eqs. (1, 2, 3) can be written as the augmented system

$$\begin{aligned} \dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{D}w(t) \\ z(t) &= \tilde{E}_\infty \tilde{x} \end{aligned} \tag{6}$$

where

$$\tilde{x} = \begin{pmatrix} x \\ x_c \end{pmatrix}$$

Using this notation and under the condition that $\tilde{A}$ is asymptotically stable, for a given compensator the performance of Eq. (5) is given by

$$J(A_c, B_c, C_c) = \operatorname{tr}(\tilde{Q}\tilde{R}) \tag{7}$$

where

$$\tilde{R} = \begin{pmatrix} R_1 & 0 \\ 0 & C_c^T R_2 C_c \end{pmatrix}$$

and $\tilde{Q}$ satisfies the Lyapunov equation

$$\tilde{A}\tilde{Q} + \tilde{Q}\tilde{A}^T + \tilde{V} = 0 \tag{8}$$

with symmetric positive semidefinite $V_1 = D_1 D^T{}_1$, symmetric positive definite $V_2 = D_2 D^T{}_2$, and

$$\tilde{V} = \begin{pmatrix} V_1 & 0 \\ 0 & B_c V_2 B_c^T \end{pmatrix}$$

Lemma 1 1. *Let (A_c, B_c, C_c) be given and assume there exists $\hat{Q} \in \boldsymbol{R}^{\tilde{n}\times\tilde{n}}$ satisfying*

$$\mathcal{Q} \text{ is symmetric and nonnegative definite} \qquad (9)$$

and

$$\tilde{A}\mathcal{Q} + \mathcal{Q}\tilde{A}^T + \gamma^{-2}\mathcal{Q}\tilde{R}_\infty\mathcal{Q} + \tilde{V} = 0 \qquad (10)$$

where

$$\tilde{R}_\infty = \begin{pmatrix} R_{1\infty} & 0 \\ 0 & C_c^T R_{2\infty} C_c \end{pmatrix}$$

$R_{1\infty} = E^T{}_{1\infty} E_{1\infty}$, and $R_{2\infty} = E^T{}_{2\infty} E_{2\infty}$ are symmetric positive semidefinite matrices. Then

$$(\tilde{A}, \tilde{D}) \text{ is stabilizable} \qquad (11)$$

if and only if

$$\tilde{A} \text{ is asymptotically stable}$$

In this case

$$\|H(s)\|_\infty \leq \gamma$$
$$\tilde{Q} \leq \mathcal{Q} \qquad (\mathcal{Q} - \tilde{Q} \text{ is nonnegative definite})$$

$$\operatorname{tr} \tilde{Q}\tilde{R} \equiv J(A_c, B_c, C_c) \leq \mathcal{J}(A_c, B_c, C_c) \equiv \operatorname{tr} \mathcal{Q}\tilde{R}$$

Hence the satisfaction of Eqs. (9) and (10) along with the generic condition of Eq. (11) leads to: (1) closed-loop stability; (2) prespecified H^∞ attenuation; and (3) an upper bound for the H^2 performance criterion.

The auxiliary minimization problem is to determine (A_c, B_c, C_c) that minimizes J(A_c, B_c, C_c) and thus provides a bound for the actual H^2 criterion J(A_c, B_c, C_c).

(A_c, B_c, C_c, $\hat{Q}$) is restricted to the open set $S \equiv \{(A_c, B_c, C_c, \hat{Q})$: $\tilde{A}$ and $\tilde{A} + \gamma^{-2}\hat{Q}\tilde{R}$ are asymptotically stable, $\hat{Q}$ is symmetric positive definite, and (A_c, B_c, C_c) is controllable and observable$\}$.

Note that if $\tilde{A}$ and $\tilde{A} + \gamma^{-2}\hat{Q}\tilde{R}$ are asymptotically stable, Eq. (10) has a unique positive definite solution $\hat{Q}$. The condition on $\hat{Q}$ in the set S is stated for clarity. However, there are no special conditions imposed in the homotopy algorithm to force $\tilde{A}$ and $\tilde{A} + \gamma^{-2}\hat{Q}\tilde{R}$ to be asymptotically stable, nor are such conditions required.

Practical applications often lead to large dense systems of nonlinear equations which are time-consuming to solve on a serial computer. For these systems, parallel processing may be the only feasible means to achieving solution algorithms with acceptable speed. One economical way of achieving parallelism is to utilize the aggregate power of a network of heterogeneous serial computers. In industrial environments where interactive design is often the practice, the parallel code can be easily incorporated into interactive software such as MATLAB or Mathematica with proper setup of the network computers. To the engineering users the design environment is identical. However, the parallel computations are faster.

The most expensive part of the H^2/H^∞ homotopy algorithm is the computation of the Jacobian matrix, which can be parallelized easily to run across an Ethernet network with little modification of the original sequential code, and which has relatively large task granularity. There is a trade-off between the programming effort and the speedup of the parallel program. To obtain a better speedup, other parts of the homotopy algorithm, such as finding the solution to the Riccati equations and the QR factorization (factorization of a matrix into an orthogonal matrix Q and an upper triangular matrix R) to compute the kernel of the Jacobian matrix, need to be parallelized as well.

In a later section, the homotopy algorithm for H^2/H^∞ controller synthesis is parallelized to run on a network of workstations using *PVM* (parallel virtual machine) and on an Intel Paragon parallel computer, under the philosophy that as few changes as possible are to be made to the sequential code while achieving an acceptable speedup. The parallelized computation is that of the Jacobian matrix, which is carried out in the master-slave paradigm by functional parallelism, that is, each machine computes a different column of the Jacobian matrix with its own data. Unless the Riccati equation solver is parallelized, there is a large amount of data needed for each slave process at each step of the homotopy algorithm. To avoid sending too many large messages across the network or among different nodes on the Intel Paragon, all slave processes repeat part of the computation done by the master process, which therefore decreases the speedup of the parallel computation.

The speedups of the parallel code are compared as the number of workstations increases or as the number of nodes increases on an Intel Paragon and as the size of the problem varies. A reasonable speedup can be achieved using an existing network of workstations compared to that of using an expensive parallel machine, the Intel Paragon. It is demonstrated that for a large problem, the approach of using a network of workstations to achieve parallelism is feasible and practical, and provides an efficient and economical computational method to parallelize a homotopy based algorithm for H^2/H^∞ controller synthesis in a workstation-based interactive design environment.

Riccati Equations

Equation (10) is a Riccati equation. In the numerical algorithm described in the later sections, Riccati Eq. (10) needs to be solved. Some of the known results about Riccati equations are summarized next.

A generalized algebraic Riccati equation can be written as

$$A^T XW + W^T XA - W^T XRXW + V = 0 \qquad (12)$$

where X is the unknown matrix, A, W, R, and V are real square matrices, and V and R are also assumed to be symmetric, with R being positive semidefinite, and W being nonsingular. For some of the applications, V is also assumed to be positive semidefinite.

Since Riccati equations are central to modern control analysis and synthesis, their theoretical properties have been thoroughly studied (15,16,17,18). Conditions that guarantee the existence of a unique symmetric solution may be also found in (19,20,21,22). Several numerical solution techniques have been developed for Riccati equations including eigenvalue methods (23,24,25,26,27,28,29,30), the Chandrasekhar algorithm (31, 32,33), and the matrix sign function technique (34). Software for Riccati equations is widely available and is included in numerous control-design packages for MATLAB and *Mathematica.*

For the given Riccati Eq. (12), the Hamiltonian pencil associated with it is defined as

$$H - \lambda L = \begin{pmatrix} A & -R \\ -V & -A^T \end{pmatrix} - \lambda \begin{pmatrix} W & 0 \\ 0 & W^T \end{pmatrix}$$

Some useful existence conditions are summarized next.

- If V is positive semidefinite, (A, R) is stabilizable, and (A, V) is observable, then there exists a unique symmetric positive semidefinite solution X.
- If V is indefinite, (A, R) is controllable, and the Hamiltonian pencil associated with the equation has no eigenvalues on the imaginary axis, then a unique symmetric solution exists.
- If both V and R are possibly indefinite and the associated Hamiltonian pencil has no pure imaginary eigenvalues, then a unique symmetric solution exists.

The homotopy algorithms described in the following sections require the solution of Eq. (10) at each point along the homotopy curve. Therefore, efficiently solving a Riccati equation is important to achieving high computational speed for the homotopy algorithm. The Schur method and an implementation by Laub (25) is used in our code.

Homotopy Algorithm Based on Ly's Formulation

In optimizing performance with respect to stabilizing controllers of fixed order n_c, it is desirable to consider controller realizations of a specified structure. In this regard there exist a variety of realizations that involve fewer than the $n_c(n_c + m + l)$ parameters appearing in a fully populated parameterization (35,36,37,38,39, 40,41,42). However, as discussed in (35,36,38), realizations that involve a minimal number of independent parameters cannot provide a smooth, global parameterization of all *MIMO* (multiple input multiple output) systems. Specialized parameterizations are also useful for realizing transfer functions of specified classes (40).

In this article we employ the Ly, Bryson, and Cannon parameterization proposed in (42). The following result characterizes the particular class of transfer functions G realized by the Ly, Bryson, and Cannon parameterization.

Proposition 1. *Suppose that G has the minimal realization (A_c, B_c, C_c), where the matrix A_c is similar to a 2 × 2 block-diagonal matrix where each 2 × 2 block has distinct eigenvalues, violation of which will lead to ill-conditioned transformation from the given basis to the Ly form (10). There is an additional 1 × 1 block if n_c is odd. Then there exists a state space basis with respect to which G has a Ly–Bryson–Cannon realization.*

Although this parameterization does not provide a global representation of all transfer functions even in the *SISO* (single input single output) case, it does provide a generic representation which is particularly suited for parametric optimization. Although the Ly–Bryson–Cannon form implicitly assumes somewhat more than that A_c be diagonalizable, computing the two-dimensional invariant subspaces is better conditioned than computing the eigenvectors, which algorithms that assume diagonalizability attempt to do. When the transformation to the Ly–Bryson–Cannon form is not ill conditioned, this particular representation turns out to be very efficient computationally.

Ly et al. (42) introduced a canonical form with $n_cm + n_cl$ parameters. The compensator is represented with respect to a basis such that A_c is a 2 × 2 block-diagonal matrix (2 × 2 blocks with an additional 1 × 1 block if n_c is odd) with 2 × 2 blocks in the form

$$\begin{pmatrix} 0 & 1 \\ * & * \end{pmatrix}$$

B_c is a full matrix, and

$$C_c = \big((C_c)_1 \quad (C_c)_2 \cdots (C_c)_r\big)$$

where

$$(C_c)_i = \begin{pmatrix} 1 & * & \cdots & * \\ 0 & * & \cdots & * \end{pmatrix}^T$$

$(C_c)_r = (1 * \cdots *)^T$ if n_c is odd.

It is assumed that (A_c, B_c, C_c) is in Ly's form. Let $\mathscr{I}$ be the set of indices of those elements of A_c which are parameters, that is

$$\mathscr{I} \equiv \{(2, 1), (2, 2), \ldots, (n_c, n_c)\}$$

To optimize $J(A_c, B_c, C_c)$ over the open set S under the constraint that symmetric positive definite $\hat{Q}$ satisfies Eq. (10), and (A_c, B_c, C_c) is in Ly's form, the following Lagrangian is formed:

$$\mathscr{L}(A_c, B_c, C_c, \mathscr{P}, \mathscr{Q}) \equiv \operatorname{tr}[\mathscr{Q}\tilde{R} + (\tilde{A}\mathscr{Q} + \mathscr{Q}\tilde{A}^T + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty\mathscr{Q} + \tilde{V})\mathscr{P}]$$

where $P \in \mathbf{R}^{\tilde{n}\times\tilde{n}}$ is a Lagrange multiplier. Setting $\partial\mathscr{L}/\partial\hat{Q} = 0$ yields

$$0 = (\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)^T\mathscr{P} + \mathscr{P}(\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty) + \tilde{R} \qquad (13)$$

Partition $\hat{Q}, P \in \mathbf{R}^{\tilde{n}\times\tilde{n}}$ as

$$\mathscr{Q} = \begin{pmatrix} \mathscr{Q}_1 & \mathscr{Q}_{12} \\ \mathscr{Q}_{12}^T & \mathscr{Q}_2 \end{pmatrix}, \qquad \mathscr{P} = \begin{pmatrix} \mathscr{P}_1 & \mathscr{P}_{12} \\ \mathscr{P}_{12}^T & \mathscr{P}_2 \end{pmatrix} \qquad (14)$$

The partial derivatives of $\mathscr{L}$ can be computed as

$$\begin{aligned}
\frac{\partial\mathscr{L}}{\partial(A_c)_{ij}} &= 2(\mathscr{P}_{12}^T\mathscr{Q}_{12} + \mathscr{P}_2\mathscr{Q}_2)_{ij}, \quad (i, j) \in \mathscr{I} \\
\frac{\partial\mathscr{L}}{\partial B_c} &= 2\mathscr{P}_2 B_c V_2 + 2(\mathscr{P}_{12}^T\mathscr{Q}_1 + \mathscr{P}_2\mathscr{Q}_{12}^T)C^T \\
\frac{\partial\mathscr{L}}{\partial(C_c)_{ij}} &= \big(2R_2C_c\mathscr{Q}_2 + 2B^T(\mathscr{P}_1\mathscr{Q}_{12} + \mathscr{P}_{12}\mathscr{Q}_2) + \gamma^{-2}R_{2\infty}C_c \\
&\quad [(\mathscr{Q}_{12}^T\mathscr{P}_1 + \mathscr{Q}_2\mathscr{P}_{12}^T)\mathscr{Q}_{12}(\mathscr{Q}_{12}^T\mathscr{P}_{12} + \mathscr{Q}_2\mathscr{P}_2)\mathscr{Q}_2]\big)_{ij} \quad i > 1
\end{aligned}$$

Let A_f, B_f, C_f, γ_f, R_{1f}, R_{2f}, $R_{1\infty f}$, $R_{2\infty f}$, V_{1f}, and V_{2f} denote A, B, C, λ, R_1, R_2, $R_{1\infty}$, $R_{2\infty}$, V_1, and V_2 in the last expression and define $A(\lambda)$, $B(\lambda)$, $C(\lambda)$, $\gamma(\lambda)$, $R_1(\lambda)$, $R_2(\lambda)$, $R_{1\infty}(\lambda)$, $R_{2\infty}(\lambda)$, $V_1(\lambda)$, $V_2(\lambda)$ as

$$\begin{aligned}
A(\lambda) &= A_0 + \lambda(A_f - A_0), \quad B(\lambda) = B_0 + \lambda(B_f - B_0)\\
C(\lambda) &= C_0 + \lambda(C_f - C_0), \quad \gamma(\lambda) = \gamma_0 + \lambda(\gamma_f - \gamma_0)\\
R_1(\lambda) &= R_{1,0} + \lambda(R_{1f} - R_{1,0}), \quad R_2(\lambda) = R_{2,0} + \lambda(R_{2f} - R_{2,0})\\
R_1\infty(\lambda) &= R_{1\infty,0} + \lambda(R_{1\infty f} - R_{1\infty,0}),\\
R_{2\infty}(\lambda) &= R_{2\infty,0} + \lambda(R_{2\infty f} - R_{2\infty,0})\\
V_1(\lambda) &= V_{1,0} + \lambda(V_{1f} - V_{1,0}), \quad V_2(\lambda) = V_{2,0} + \lambda(V_{2f} - V_{2,0})
\end{aligned} \tag{15}$$

and denote them by A, B, C, γ, R_1, R_2, $R_{1\infty}$, $R_{2\infty}$, V_1, and V_2 respectively in the following. Let

$$\begin{aligned}
H_{A_c}(\theta,\lambda) &= \frac{1}{2}\frac{\partial \mathscr{L}}{\partial A_c} = \mathscr{P}_{12}^T\mathscr{Q}_{12} + \mathscr{P}_2\mathscr{Q}_2\\
H_{B_c}(\theta,\lambda) &= \frac{1}{2}\frac{\partial \mathscr{L}}{\partial \mathscr{B}_c} = \mathscr{P}_2 B_c V_2 + (\mathscr{P}_{12}^T\mathscr{Q}_1 + \mathscr{P}_2\mathscr{Q}_{12}^T)C^T\\
H_{C_c}(\theta,\lambda) &= \frac{1}{2}\frac{\partial \mathscr{L}}{\partial C_c} = R_2 C_c \mathscr{Q}_2 + B^T(\mathscr{P}_1\mathscr{Q}_{12} + \mathscr{P}_{12}\mathscr{Q}_2)\\
&\quad + \frac{1}{2}\gamma^{-2}R_{2\infty}C_c[(\mathscr{Q}_{12}^T\mathscr{P}_1 + \mathscr{Q}_2\mathscr{P}_{12}^T)\mathscr{Q}_{12}\\
&\quad + (\mathscr{Q}_{12}^T\mathscr{P}_{12} + \mathscr{Q}_2\mathscr{P}_2)\mathscr{Q}_2]
\end{aligned} \tag{16}$$

where in H_{Ac} only those elements corresponding to the parameter elements of A_c are of interest and

$$\theta \equiv \begin{pmatrix} (A_c)_{\mathscr{I}} \\ \mathbf{Vec}\,(B_c) \\ \mathbf{Vec}\,(C_c)_{\mathscr{T}}. \end{pmatrix} \tag{17}$$

denotes the independent variables, $\hat{Q}$ and P satisfy respectively Eqs. (10) and (13), $(A_c)_{\mathscr{I}}$ is a vector consisting of those elements in A_c with indices in the set I, that is

$$(A_c)_{\mathscr{I}} = \left((A_c)_{21}, (A_c)_{22}, \ldots, (A_c)_{n_c n_c}\right)^T$$

and $(C_c)\,T$ is the matrix obtained from rows $P = \{2, \ldots, m\}$ of C_c. Vec(P) for a matrix $P \in \mathbf{R}^{p\times q}$ is the concatenation of its columns:

$$\mathbf{Vec}(P) \equiv \begin{pmatrix} P_{\cdot 1} \\ P_{\cdot 2} \\ \vdots \\ P_{\cdot q} \end{pmatrix} \in \mathbf{R}^{pq}$$

The most natural homotopy map, given the embeddings in Eq. (15), is defined as

$$\rho(\theta, \lambda) = \begin{pmatrix} [H_{A_c}(\theta, \lambda)]_{\mathscr{S}} \\ \mathbf{Vec}\,[H_{B_c}(\theta, \lambda)] \\ \mathbf{Vec}\,[H_{C_c}(\theta, \lambda)]_{\mathscr{S}}. \end{pmatrix} \tag{18}$$

and its Jacobian matrix is

$$D\rho(\theta, \lambda) = \big(D_\theta \rho(\theta, \lambda), D_\lambda \rho(\theta, \lambda)\big)$$

In practice it may be difficult to find the initial point θ_0 such that $\rho(\theta_0, 0) = 0$. A somewhat more artificial (lacking a physical interpretation) homotopy then, letting θ_0 be the chosen initial point, is the Newton homotopy map defined as

$$\overline{\rho}(\theta, \lambda) = \rho(\theta, \lambda) - (1 - \lambda)\rho(\theta_0, 0)$$

which will give rise to an extra term $\rho(\theta_0,\ 0)$ in $D_\lambda \bar{\rho}(\theta,\lambda)$. To guarantee a full rank Jacobian matrix along the whole homotopy zero curve except possibly at the solution corresponding to $\lambda = 1$, define the homotopy map to be

$$\hat{\rho}(\theta, \lambda) = \lambda\rho(\theta, \lambda) + (1 - \lambda)(\theta - \theta_0) \tag{19}$$

The Jacobian matrix of $\bar{\rho}$ is given by

$$\begin{aligned} D\hat{\rho}(\theta, \lambda) = \big(&\lambda D_\theta \rho(\theta, \lambda) + (1 - \lambda)I,\ \rho(\theta, \lambda) \\ &+ \lambda D_\lambda \rho(\theta, \lambda) - (\theta - \theta_0)\big) \end{aligned}$$

In the following, the homotopy map of Eq. (18) is assumed for the full-order problem and Eq. (19) is assumed for the reduced-order case since the reduced-order initialization scheme produces a singular starting point if Eq. (18) is used. Define

$$\begin{aligned} \hat{H}_{A_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) =& \mathscr{P}_{12}^{T(j)}\mathscr{Q}_{12} + \mathscr{P}_{12}^{T}\mathscr{Q}_{12}^{(j)} + \mathscr{P}_{2}^{(j)}\mathscr{Q}_{2} + \mathscr{P}_{2}\mathscr{Q}_{2}^{(j)} \\ \hat{H}_{B_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) =& \mathscr{P}_{2}^{(j)}B_cV_2 + (\mathscr{P}_{12}^{T(j)}\mathscr{Q}_{1} + \mathscr{P}_{12}^{T}\mathscr{Q}_{1}^{(j)} \\ &+ \mathscr{P}_{2}^{(j)}\mathscr{Q}_{12}^{T} + \mathscr{P}_{2}\mathscr{Q}_{12}^{T(j)})C^T \\ \hat{H}_{C_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) =& R_2C_c\mathscr{Q}_{2}^{(j)} + B^T(\mathscr{P}_{1}^{(j)}\mathscr{Q}_{12} + \mathscr{P}_{1}\mathscr{Q}_{12}^{(j)} \\ &+ \mathscr{P}_{12}^{(j)}\mathscr{Q}_{2} + \mathscr{P}_{12}\mathscr{Q}_{2}^{(j)}) \\ &+ \frac{\gamma^{-2}}{2}R_{2\infty}C_c\big[(\mathscr{Q}_{12}^{T(j)}\mathscr{P}_{1} + \mathscr{Q}_{12}^{T}\mathscr{P}_{1}^{(j)} \\ &+ \mathscr{Q}_{2}^{(j)}\mathscr{P}_{12}^{T} + \mathscr{Q}_{2}\mathscr{P}_{12}^{T(j)})\mathscr{Q}_{12} \\ &+ (\mathscr{Q}_{12}^{T(j)}\mathscr{P}_{12} + \mathscr{Q}_{12}^{T}\mathscr{P}_{12}^{(j)} + \mathscr{Q}_{2}^{(j)}\mathscr{P}_{2} \\ &+ \mathscr{Q}_{2}\mathscr{P}_{2}^{(j)})\mathscr{Q}_{2} + (\mathscr{Q}_{12}^{T}\mathscr{P}_{1} \\ &+ \mathscr{Q}_{2}\mathscr{P}_{12}^{T})\mathscr{Q}_{12}^{(j)} + (\mathscr{Q}_{12}^{T}\mathscr{P}_{12} + \mathscr{Q}_{2}\mathscr{P}_{2})\mathscr{Q}_{2}^{(j)}\big] \end{aligned} \tag{20}$$

where the superscript (j) means $\partial/\partial\theta_j$, Using these definitions, we have for $\theta_j = (A_c)_{kl}$, where $(k,l) \in \mathscr{I}$,

$$\begin{aligned} \frac{\partial H_{A_c}}{\partial (A_c)_{kl}} &= \hat{H}_{A_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) \\ \frac{\partial H_{B_c}}{\partial (A_c)_{kl}} &= \hat{H}_{B_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) \\ \frac{\partial H_{C_c}}{\partial (A_c)_{kl}} &= \hat{H}_{C_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) \end{aligned} \tag{21}$$

for $\theta_j = (B_c)_{kl}$,

$$\begin{aligned} \frac{\partial H_{A_c}}{\partial (B_c)_{kl}} &= \hat{H}_{A_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) \\ \frac{\partial H_{B_c}}{\partial (B_c)_{kl}} &= \hat{H}_{B_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) + \mathscr{P}_2 E^{(k,l)} V_2 \\ \frac{\partial H_{C_c}}{\partial (B_c)_{kl}} &= \hat{H}_{C_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) \end{aligned} \tag{22}$$

and for $\theta_j = (C_c)_{kl}$, where $k > 1$,

$$\begin{aligned} \frac{\partial H_{A_c}}{\partial (C_c)_{kl}} &= \hat{H}_{A_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) \\ \frac{\partial H_{B_c}}{\partial (C_c)_{kl}} &= \hat{H}_{B_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) \\ \frac{\partial H_{C_c}}{\partial (C_c)_{kl}} &= \hat{H}_{C_c}(\mathscr{P}^{(j)}, \mathscr{Q}^{(j)}) + R_2 E^{(k,l)} \mathscr{Q}_2 + \frac{\gamma^{-2}}{2} R_{2\infty} E^{(k,l)} Y \end{aligned} \tag{23}$$

where

$$Y = (\mathscr{Q}_{12}^T \mathscr{P}_1 + \mathscr{Q}_2 \mathscr{P}_{12}^T)\mathscr{Q}_{12} + (\mathscr{Q}_{12}^T \mathscr{P}_{12} + \mathscr{Q}_2 \mathscr{P}_2)\mathscr{Q}_2 \tag{24}$$

and $E^{(k,l)}$ is a matrix of the appropriate dimension of which the only nonzero element is $e_{kl} = 1$. $P^{(j)}$ and $\hat{Q}^{(j)}$ can be obtained by solving the Lyapunov equations

$$\begin{aligned} 0 &= (\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)\mathscr{Q}^{(j)} + \mathscr{Q}^{(j)}(\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)^T + \tilde{V}^{(j)} \\ &\quad + \tilde{A}^{(j)}\mathscr{Q} + \mathscr{Q}\tilde{A}^{T(j)} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty^{(j)}\mathscr{Q} \\ 0 &= (\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)^T\mathscr{P}^{(j)} + \mathscr{P}^{(j)}(\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty) + \tilde{R}^{(j)} \\ &\quad + (\tilde{A}^{(j)} + \gamma^{-2}\mathscr{Q}^{(j)}\tilde{R}_\infty \\ &\quad + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty^{(j)})^T\mathscr{P} + \mathscr{P}(\tilde{A}^{(j)} + \gamma^{-2}\mathscr{Q}^{(j)}\tilde{R}_\infty + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty^{(j)}) \end{aligned} \tag{25}$$

Similarly for λ, using a dot to denote ∂/∂λ,

$$
\begin{aligned}
\frac{\partial H_{A_c}}{\partial \lambda} &= \hat{H}_{A_c}(\dot{\mathscr{P}}, \dot{\mathscr{Q}}) \\
\frac{\partial H_{B_c}}{\partial \lambda} &= \hat{H}_{B_c}(\dot{\mathscr{P}}, \dot{\mathscr{Q}}) + \mathscr{P}_2 B_c \dot{V}_2 + (\mathscr{P}_{12}^T \mathscr{Q}_1 + \mathscr{P}_2 \mathscr{Q}_{12}^T)\dot{C}^T \\
\frac{\partial H_{C_c}}{\partial \lambda} &= \hat{H}_{C_c}(\dot{\mathscr{P}}, \dot{\mathscr{Q}}) + R_2 C_c \mathscr{Q}_2 + \dot{B}^T(\mathscr{P}_1 \mathscr{Q}_{12} + \mathscr{P}_{12} \mathscr{Q}_2) \\
&\quad + \frac{1}{2}\gamma^{-2}\dot{R}_{2\infty} C_c Y - \gamma^{-3}\dot{\gamma} R_{2\infty} C_c Y
\end{aligned}
\tag{26}
$$

where $\dot{P}$ and $\dot{Q}$ are obtained by solving

$$
\begin{aligned}
0 &= (\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)\dot{\mathscr{Q}} + \dot{\mathscr{Q}}(\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)^T + \dot{\tilde{V}} + \dot{\tilde{A}}\mathscr{Q} \\
&\quad + \mathscr{Q}\dot{\tilde{A}}^T + \gamma^{-2}\mathscr{Q}\dot{\tilde{R}}_\infty \mathscr{Q} - 2\gamma^{-3}\dot{\gamma}\mathscr{Q}\tilde{R}_\infty \mathscr{Q} \\
0 &= (\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)^T \dot{\mathscr{P}} + \dot{\mathscr{P}}(\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty) + \dot{\tilde{R}} \\
&\quad + (\dot{\tilde{A}} + \gamma^{-2}\dot{\mathscr{Q}}\tilde{R}_\infty + \gamma^{-2}\mathscr{Q}\dot{\tilde{R}} + \infty - 2\gamma^{-3}\dot{\gamma}\mathscr{Q}\tilde{R}_\infty)^T \mathscr{P} \\
&\quad + \mathscr{P}(\dot{\tilde{A}} + \gamma^{-2}\dot{\mathscr{Q}}\tilde{R}_\infty + \gamma^{-2}\mathscr{Q}\dot{\tilde{R}}_\infty - 2\gamma^{-3}\dot{\gamma}\mathscr{Q}\tilde{R}_\infty)
\end{aligned}
\tag{27}
$$

Homotopy Algorithm Based on Overparametrization Formulation

The parametrization in the previous section, the Ly form, is minimal in the sense that it uses the minimum possible number of parameters, $n_c(m + l)$, to describe the controller. However, it also assumes a particular structure for the controller that may not be satisfied by the optimal controller, or even if it is, that structure may be ill conditioned near the optimum. The formulation in this section makes no assumptions whatsoever on the controller structure, treating all the components of (A_c, B_c, C_c) as independent variables.

To optimize $T(A_c, B_c, C_c)$ over the open set S under the constraint that symmetric positive definite $\hat{Q}$ satisfies Eq. (10), the following Lagrangian is formed:

$$
\mathscr{L}(A_c, B_c, C_c, \mathscr{P}, \mathscr{Q}) \equiv \mathrm{tr}\left[\mathscr{Q}\tilde{R} + (\tilde{A}\mathscr{Q} + \mathscr{Q}\tilde{A}^T + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty \mathscr{Q} + \tilde{V})\mathscr{P}\right]
$$

where$P \in \mathbf{R}^{\tilde{n}\times\tilde{n}}$ is a Lagrange multiplier. Setting $\partial\mathscr{L}/\partial\hat{Q} = 0$ yields Eq. (13). Partition $\hat{Q}$,$P \in \mathbf{R}^{\tilde{n}\times\tilde{n}}$ as in Eq. (14). The partial derivatives of $\mathscr{L}$ can be computed as

$$
\begin{aligned}
\frac{\partial \mathscr{L}}{\partial A_c} &= 2(\mathscr{P}_{12}^T \mathscr{Q}_{12} + \mathscr{P}_2 \mathscr{Q}_2) \\
\frac{\partial \mathscr{L}}{\partial B_c} &= 2\mathscr{P}_2 B_c V_2 + 2(\mathscr{P}_{12}^T \mathscr{Q}_1 + \mathscr{P}_2 \mathscr{Q}_{12}^T)C^T \\
\frac{\partial \mathscr{L}}{\partial C_c} &= 2R_2 C_c \mathscr{Q}_2 + 2B^T(\mathscr{P}_1 \mathscr{Q}_{12} + \mathscr{P}_{12} \mathscr{Q}_2) \\
&\quad + \gamma^{-2} R_{2\infty} C_c\left[(\mathscr{Q}_{12}^T \mathscr{P}_1 + \mathscr{Q}_2 \mathscr{P}_{12}^T)\mathscr{Q}_{12} + (\mathscr{Q}_{12}^T \mathscr{P}_{12} + \mathscr{Q}_2 \mathscr{P}_2)\mathscr{Q}_2\right]
\end{aligned}
$$

Let A_f, B_f, Cf, γ_f, R_{1f}, R_{2f}, $R_{1\infty f}$, $R_{2\infty f}$, V_{1f}, and V_{2f} denote A, B, C, λ, R_1, R_2, $R_{1\infty}$, $R_{2\infty}$, V_1, and V_2 in the last expression and define $A(\lambda)$, $B(\lambda)$, $C(\lambda)$, $\gamma(\lambda)$, $R_1(\lambda)$, $R_2(\lambda)$, $R_{1\infty}(\lambda)$, $R_{2\infty}(\lambda)$, $V_1(\lambda)$, and $V_2(\lambda)$ as in Eq. (15) and denote them by A, B, C, γ, R_1, R_2, $R_{1\infty}$, $R_{2\infty}$, V_1, and V_2 respectively in the following.

Define $H_{Ac}(\theta, \lambda)$, $H_{Bc}(\theta, \lambda)$, and $H_{Cc}(\theta, \lambda)$ as in Eq. (16) where

$$\theta \equiv \begin{pmatrix} \text{Vec}\,(A_c) \\ \text{Vec}\,(B_c) \\ \text{Vec}\,(C_c) \end{pmatrix} \tag{28}$$

denotes the independent variables, and $\hat{Q}$ and$P$satisfy respectively Eq. (10) and Eq. (13). Vec applied to a matrix is a column vector obtained by concatenating the columns of the matrix.

Define

$$\rho(\theta, \lambda) = \begin{pmatrix} \text{Vec}\,[H_{A_c}(\theta, \lambda)] \\ \text{Vec}\,[H_{B_c}(\theta, \lambda)] \\ \text{Vec}\,[H_{C_c}(\theta, \lambda)] \end{pmatrix} \tag{29}$$

whose Jacobian matrix is

$$D\rho(\theta, \lambda) = \big(D_\theta \rho(\theta, \lambda), D_\lambda \rho(\theta, \lambda)\big)$$

Note that θ in Eq. (29) has $n^2{}_c + n_c m + n_c l$ components, more than the minimal number $n_c m + n_c l$ needed. Because of this over-parametrization, the Jacobian matrix of ρ is seriously rank deficient. To remedy this severe rank deficiency, the homotopy map is defined as

$$\hat{\rho}(\theta, \lambda) = \lambda \rho(\theta, \lambda) + (1 - \lambda)(\theta - \theta_0) \tag{30}$$

which guarantees a full rank Jacobian matrix along the entire homotopy zero curve except possibly at the solution (corresponding to $\lambda = 1$). The Jacobian matrix of $\bar{\rho}$ is given by

$$\begin{aligned} &D\hat{\rho}(\theta, \lambda) \\ &\quad = \big(\lambda D_\theta \rho(\theta, \lambda) + (1 - \lambda) I,\ \rho(\theta, \lambda) + \lambda D_\lambda \rho(\theta, \lambda) - (\theta - \theta_0)\big) \end{aligned} \tag{31}$$

To find $D_\theta \rho(\theta, \lambda)$, define the auxiliary matrices $\hat{H}_{Ac}(P^{(j)}, \hat{Q}^{(j)})$, $\hat{H}_{Bc}(P^{(j)}, \hat{Q}^{(j)})$, and $\hat{H}_{Cc}(P^{(j)}, \hat{Q}^{(j)})$ as in Eq. (20). Using these definitions, we have Eqs. (21), (22), (23) for $\theta_j = (A_c)_{kl}$, $\theta_j = (B_c)_{kl}$, and $\theta_j = (C_c)_{kl}$ respectively. $P^{(j)}$ and $\hat{Q}^{(j)}$ can be obtained by solving the Lyapunov Eq. (25). Similarly we have Eq. (26) for λ and $\dot{p}$ and $\dot{Q}$ are obtained by solving Eq. (27).

Numerical Algorithms

Choose r_0, the initial γ, so that $\gamma^{-2}{}_0$ is approximately zero. The initial point $(\theta, \lambda) = (\theta_0, 0)$ is chosen so that it satisfies $\rho(\theta_0, 0) = 0$ and the triple $((A_c)_0, (B_c)_0, (C_c)_0)$ is in the respective form for each homotopy.

It is well known that the full-order LQG compensator of Eq. (2) for the plant Eq. (1) minimizing the steady-state quadratic performance functional Eq. (5) is given by:

$$A_c = A - \Sigma P - Q\bar{\Sigma} \tag{32}$$

$$B_c = QC^TV_2^{-1}, \quad C_c = -R_2^{-1}B^tP \tag{33}$$

where $\Sigma \equiv BR^{-1}{}_2B^T$, $\Sigma \equiv C^TV^{-1}{}_2C$, and P and Q are the unique, symmetric, positive semidefinite solutions respectively, of

$$\begin{aligned} 0 &= A^TP + PA + R_1 - P\Sigma P \\ 0 &= AQ + QA^T + V_1 - Q\bar{\Sigma}Q \end{aligned} \tag{34}$$

Full-Order Initialization. The initial point for the full-order problem can be chosen as follows:

(1) Solve for Q and P from Eq. (34) and obtain $((\hat{A}_c)_0, (\hat{B}_c)_0, (\hat{C}_c)_0)$ from Eqs. (32) and (33).
(2) Transform the triple $((\hat{A}_c)_0, (\hat{B}_c)_0, (\hat{C}_c)_0)$ to Ly's form for the Ly form homotopy, and build θ_0 as described in Eq. (17) and Eq. (28) for the respective homotopies.

Reduced-Order Initization. The initialization scheme for the reduced-order problem is more complicated since a closed form expression for the reduced-order H^2 LQG compensator does not exist. For a given system $(\bar{A}, \bar{B}, \bar{C})$, and matrices $\bar{R}_1, \bar{R}_2, \bar{R}_{1\infty}, \bar{R}_{2\infty}, \bar{V}_1$, and $\bar{V}_2$, the reduced order starting point is chosen using a method in (12) which can be summarized as:

(1) Compute the real Schur decomposition of $\bar{A}$ so that $\bar{A} = UAU^t$,

$$A \equiv \begin{pmatrix} A_1 & 0 \\ A_{21} & A_2 \end{pmatrix}$$

where $A_1 \in R^{n_c \times n_c}$, and transform $\bar{B}, \bar{C}, \bar{R}_1, \bar{V}_1, \bar{R}_{1\infty}$ so that $B = U\bar{B}$, $C = \bar{C}U^t$, $R_1 = U\bar{R}_1U^t$, $V_1 = U\bar{V}_1U^t$, $R_{1\infty} = U\bar{R}_{1\infty}U^t$ and let $R_2 = \bar{R}_2$, $V_2 = \bar{V}_2$, $R_{2\infty} = \bar{R}_{2\infty}$.
(2) If A is not asymptotically stable, modify either diagonal elements or 2 × 2 diagonal blocks of A so that it is asymptotically stable and call this modified matrix A_0. Note that this step can always be done easily. For example, a diagonal matrix can be added to A_0 until the sum is asymptotically stable.
(3) Take $B_0 = B$, $C_0 = C$, $R_{2,0} = R_{2f} \equiv R_2$, $R_{2\infty,0} = R_{2\infty\mathrm{f}} \equiv R_{2\infty}$, $V_{1,0} = V_{1f} \equiv V_1$, $V_{2,0} = \beta V_{2f} \equiv \beta V_2$, $\beta \gg 0$, and

$$R_{1,0} = \begin{pmatrix} (R_1)_1 & 0 \\ 0 & 0 \end{pmatrix}$$

where $(R_1)_1$ is the leading $n_c \times n_c$ block of $R_{1f} \equiv R_1$.
(4) Solve

$$\begin{aligned} 0 &= A_0^TP + PA_0 + R_{1,0} - P\Sigma_0P \\ 0 &= A_0Q + QA_0^T + V_{1,0} - Q\bar{\Sigma}_0Q \end{aligned}$$

for symmetric, positive semidefinite P and Q, where $\Sigma_0 \equiv B_0 R^{-1}{}_{2,0} B^T{}_0$, and $\bar{\Sigma}_0 \equiv C^T{}_0 V^{-1}{}_{2,0} C_0$.

(5) Obtain (A_c, B_c, C_c) from $A_c = A_0 - \Sigma_0 P - Q\bar{\Sigma}_0$, $B_c = QC^T{}_0 V^{-1}{}_{2,0}$, $C_c = -R^{-1}{}_{2,0} B^T{}_0 P$.

(6) Solve

$$0 = (A_0 - Q\bar{\Sigma}_0)^T \hat{P} + \hat{P}(A_0 - Q\bar{\Sigma}_0) + P\Sigma_0 P$$
$$0 = (A_0 - \Sigma_0 P)\hat{Q} + \hat{Q}(A_0 - \Sigma_0 P)^T + Q\bar{\Sigma}_0 Q$$

for symmetric, positive semidefinite $\hat{P}$ and $\hat{Q}$.

Following (43), obtain the reduced-order compensator starting point from (A_c, B_c, C_c), $\hat{P}$, $\hat{Q}$, and n_c, as follows:

(1) Compute the Cholesky decomposition of (assumed positive definite) $\hat{P}$ and $\hat{Q}$,that is $\hat{P} = L\ \hat{P}\ L^T\ \hat{P}$, $\hat{Q} = L\ \hat{Q}\ L^T\ \hat{Q}$.

(2) Compute the singular value decomposition of $L^T\ \hat{P}\ L\ \hat{Q}$, that is, $L^T\ \hat{P}\ L\ \hat{Q} = U\Omega V^T$.

(3) Let $T = L\ \hat{Q}\ V\Omega^{-1/2}$, $T^{-1} = \Omega^{-1/2} U^T L^T\ \hat{P}$.

(4) Let $\bar{A}_c = T^{-1} A_c T$, $\bar{B}_c = T^{-1} B_c$, and $\bar{C}_c = C_c T$ so that

$$\bar{A}_c = {}^{n_c}\{\begin{pmatrix} \overbrace{(\bar{A}_c)_1}^{n_c} & (\bar{A}_c)_{12} \\ (\bar{A}_c)_{21} & (\bar{A}_c)_2 \end{pmatrix} \quad \bar{B}_c = {}^{n_c}\{\begin{pmatrix} (\bar{B}_c)_1 \\ (\bar{B}_c)_2 \end{pmatrix}$$
$$\bar{C}_c = \begin{pmatrix} \overbrace{(\bar{C}_c)_1}^{n_c} & (\bar{C}_c)_2 \end{pmatrix}$$

The starting point θ_0 for the reduced order problem is chosen using $((\bar{A}_c)_1, \bar{B}_c)_1, (\bar{C}_c)_1)$, with the construction in Eq. (17) and Eq. (28) for the respective homotopies.

The main idea of choosing the initial point is to find an approximate H^2 solution as the initial point so that the corresponding γ is very big. As λ increases to 1, γ goes to the given value. Computationally, if choosing $\gamma_0 = 10^5$ and $\gamma_0 = 10^6$ lead to the same solution, then the initial γ_0 can be chosen as 10^5. If γ is too small for the existence of a solution, the Riccati solver fails. In other words, it is impossible to obtain a symmetric and positive definite solution $\hat{Q}$ from Eq. (10) in this situation.

Homotopy Zero Curve Tracking. Once the initial point is chosen, the rest of the computation is as follows:

(1) Set $\lambda := 0$, $\theta := \theta_0$.

(2) Calculate $\tilde{R}$, $\tilde{R}_\infty$, $\tilde{V}$, and compute $\hat{Q}$ and$P$according to Eqs. (10) and (13).

(3) Evaluate the homotopy map $\rho(\theta, \lambda)$ or $\hat{\rho}(\theta, \lambda)$ and the Jacobian of the homotopy map $\mathrm{D}\rho(\theta, \lambda)$ or $\mathrm{D}\hat{\rho}(\theta, \lambda)$.

(4) Predict the next point $Z^{(0)} = (\theta^{(0)}, \lambda^{(0)})$ on the homotopy zero curve using, for example, a Hermite cubic interpolant.

(5) For k := 0, 1, 2, ··· until convergence do

$$Z^{(k+1)} = Z^{(k)} - [D\rho(Z^{(k)})]^{\dagger} \rho(Z^{(k)})$$

where $[D\rho(Z)]^\dagger$ is the Moore–Penrose inverse of $D\rho(Z)$. Let $(\theta_1, \lambda_1) = \lim_{k\to\infty} Z^{(k)}$.

(6) If $\lambda_1 < 1$, then set $\theta := \theta_1$, $\lambda := \lambda_1$, and so to step 2.

(7) If $\lambda_1 \geq 1$, compute the solution $\bar{\theta}$ at $\lambda = 1$.

For the over-parametrization formulation homotopy, because of the singularity at $\lambda = 1$, step 7 is replaced by:

(1) If $\lambda_1 \geq 1$, use the last point $(\tilde{\lambda}, \tilde{\lambda})$ with $\tilde{\lambda} < 1$ to redefine the homotopy map with $\theta_0 = \tilde{\lambda}$.

(2) Redo steps 1–6 until $\lambda \geq 1$.

(3) Use Hermite polynomial interpolation to obtain the solution at $\lambda = 1$.

The Distributed Homotopy Algorithm

In the preceding algorithm, step 2 involves solving one Riccati equation and one Lyapunov equation. The Riccati equation is solved using Laub's Schur method (25). The algorithm of Bartels and Stewart (44) is applied to solve the Lyapunov equation. Although both algorithms are $O((n + n_c)^3)$, the Riccati equation, being more complicated, takes much more *CPU* (central processing unit) time to solve. Once $\hat{Q}$ and$P$are obtained, the homotopy map $\bar{\rho}$ is formed by matrix multiplication operations.

The major part of the computation in step 3 is that of the Jacobian matrix. The number of variables including λ in this formulation is $n_c(m + l) + 1$. Each column of the Jacobian matrix corresponds to the derivative of the homotopy map with respect to one variable and requires the solution of two Lyapunov equations (9). Therefore, the time complexity of the Jacobian matrix computation is $O(n_c(m + l)(n + n_c)^3)$. The Bartels and Stewart algorithm finds the real Schur form of $\tilde{A}$ or $\tilde{A}^T$ depending on the Lyapunov equation. At each step along the homotopy path unnecessary factorization can be avoided if the previous factorization results from the computation of $\bar{\rho}$ and $D_\lambda\hat{\rho}$ are used.

Our goal of distributed computation is to make use of the existing code and to achieve reasonable parallel efficiency economically. The only part of the algorithm that is parallelized is the Jacobian matrix computation in step 3. To utilize existing computer resources such as a network of workstations, the software package PVM (parallel virtual machine) is used to provide the distributed computing capabilities.

The parallel algorithm follows the master-slave paradigm. The master sends the index of the column of the Jacobian matrix to be computed to a slave. The slave computes the corresponding column of the Jacobian, sends the column back to the master, and waits for the next index from the master to arrive. After receiving a column of the Jacobian, the master sends another index to the idle slave. In the implementation for the Intel Paragon, asynchronous send, which sends a message without waiting for completion, is used whenever possible to speed up the communication.

When the algorithm is implemented on a network of workstations, the modification to the original sequential source code consists of three parts: the first one is to spawn slave processes and set up the communication links between the master and the slaves; the second is to extract a slave program from the original code and at the same time simplify the master program; the last is to add a mechanism to guarantee correct communication between master and slaves. The first part consists of standard PVM operations, while the second is more problem oriented. To decrease communication, each slave process repeats part of the computation of $\bar{\rho}$ and $D_\lambda\hat{\rho}$ so that $\hat{Q}$ and$P$are not sent through the network. There is no loss of efficiency since the master is also computing the same quantities. The slave program consists of mainly the original subroutines with additional code for communication.

For the implementation on the Intel Paragon, the modification of the original code is even simpler. There is no need for a separate slave program if control statements use node identification properly. The parent process, which runs on an Intel Paragon, always gets node number 0, while other nodes are numbered 1 and higher. The statement *if node_number == 0* precedes the code that is to be executed by the master, and an *else* following the previous master code will precede the code to be executed by the slave. The remaining modification to the original code is similar to the implementation using PVM. Asynchronous send is used whenever possible. A *wait* is used later when the data is needed, to ensure correct communication between the master and the slaves.

Numerical Results and Discussion

The following systems are solved by the homotopy algorithms discussed in the previous sections. The homotopy curve tracking was done with HOMPACK (14).

The first system, formulated in Ref. 45 and studied in Ref. 1, is given by

$$A = \begin{pmatrix} -0.161 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6.004 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -0.5822 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -9.9835 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -0.4073 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -3.982 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ 0.0064 \\ 0.00235 \\ 0.0713 \\ 1.0002 \\ 0.1045 \\ 0.9955 \end{pmatrix} \qquad C^T = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$E_1 = 10^{-3} \begin{pmatrix} 0 & 0 & 0 & 0 & 0.55 & 11 & 1.32 & 18 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad E_{1\infty} = E_1$$

$$E_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad E_{2\infty} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad D_1 = (B \quad 0), \quad D_2 = (0 \quad 1)$$

For a given initial (A_c, B_c, C_c), γ is lowered until a solution cannot be found anymore. The smallest γ for which a solution can still be found is $\gamma_{\min}$. For the full-order (i.e., 8th order) problem, the solutions of the auxiliary minimization problem are obtained for $\gamma \geq \gamma_{\min} \equiv 0.481$ using the Ly form homotopy approach. For $\gamma < \gamma_{\min} = 0.481$, the Riccati equation solver fails and therefore no solution can be found. In Fig. 1, $\|H(s)\|_\infty$ is plotted against J. The ratio of $\|H(s)\|_\infty$ at $\gamma = \gamma_{\min}$ to that at $\gamma = \infty$ is 0.33, which indicates that there is about 67% improvement in the H^∞ performance of the compensator with $\gamma = \gamma_{\min}$ over the compensator without the H^∞ constraint.

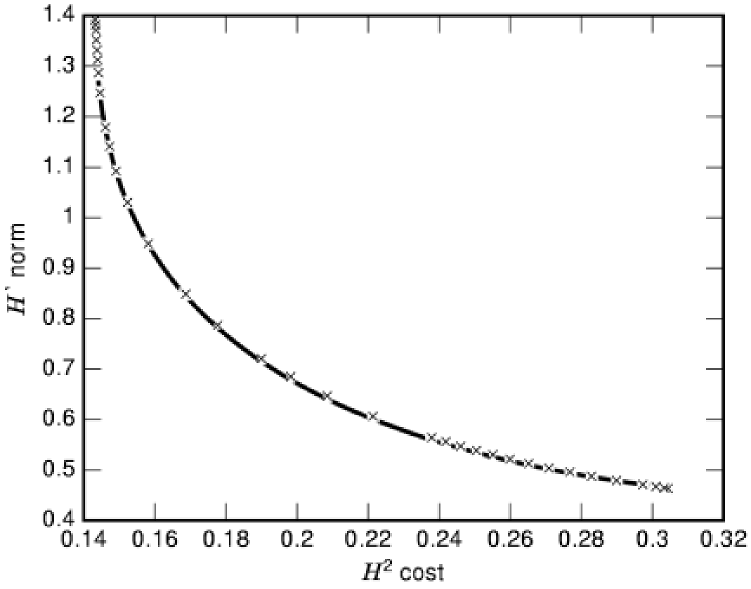

Fig. 1. $\|H(s)\|_\infty$ versus J for $n_c = n$.

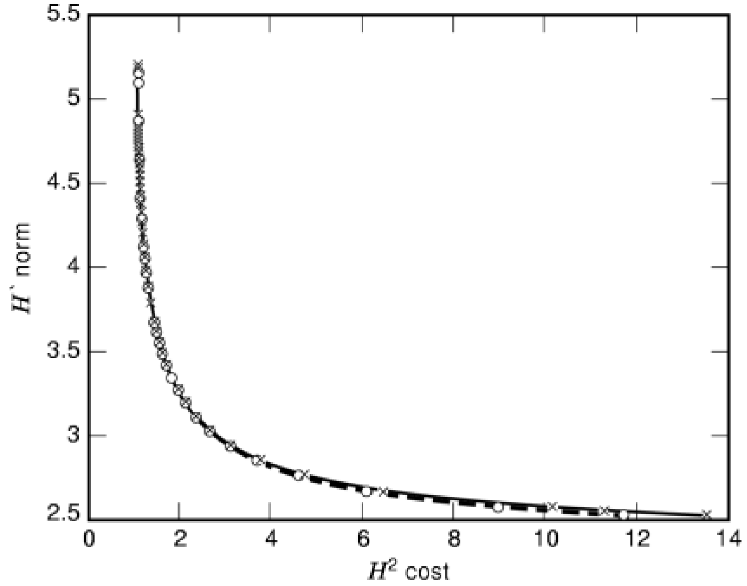

Fig. 2. $\|H(s)\|_\infty$ versus $J$for $n_c = 4$, 6.

For $n_c = 4$, 6, the solutions of the auxiliary minimization problem are obtained for $\gamma \geq 2.55$ using the Ly form homotopy approach. In Fig. 2, $\|H(s)\|_\infty$ is plotted against J for $n_c = 4$ (solid line with "x" indicating the data points) and $n_c = 6$ (dashed line with "o" indicating the data points). For both $n_c = 4$ and $n_c = 6$, the ratio of $\|H(s)\|_\infty$ at $\gamma = 2.55$ to that at $\gamma = \infty$ is 0.49, which indicates that there is about 51% improvement in the

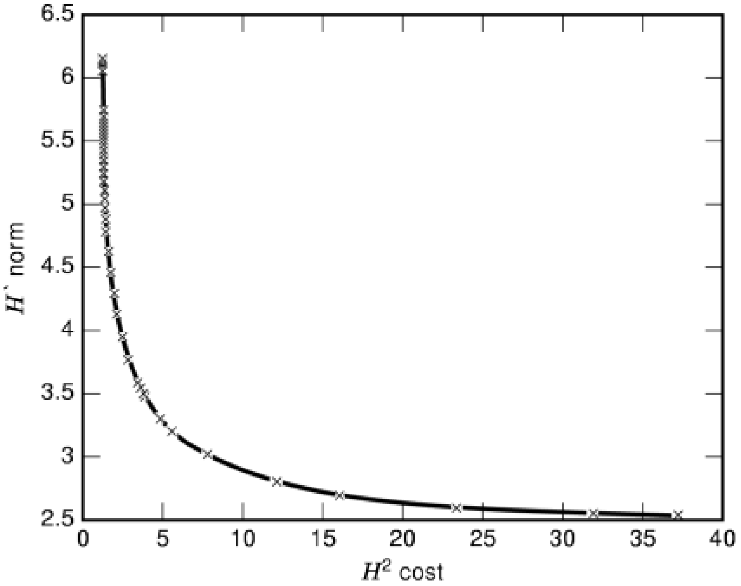

Fig. 3. $\|H(s)\|_\infty$ versus J for $n_c = 2$.

H^∞ performance of the compensator with $\gamma = 2.55$ over the compensator without the H^∞ constraint. For this example, the 4th-order and 6th-order compensators have almost the same H^2 and H^∞ performance.

For $n_c = 2$, two different sets of solutions are obtained by varying β in the initialization step. Different β correspond to different initial (A_c, B_c, C_c), and therefore different homotopy curves. In this case, these two different homotopy curves lead to different solutions which have different minimum H^∞ upper bounds $\gamma_{\min}$.

The trade-off curves are shown in Fig. 3 ($\beta = 100$) and Fig. 4 ($\beta = 1$). The first set of solutions (shown in Fig. 3) is obtained for $\gamma \geq 2.54$, while the second set (shown in Fig. 4) is obtained for $\gamma \geq 9.5$. It can be seen that the first set of solutions has lower H^2 cost and better H^∞ performance. It was verified by sampling in a neighborhood that all the points in both Figs. 3 and 4 are local minima of the auxiliary cost J.

The homotopy algorithms proposed here are examples of probability-one globally convergent homotopy methods; the matrices A_0, B_0, …, and the starting point θ_0 here play the role of the parameter vector a in the probability-one homotopy theory (13). Figure 5 shows a portion of γ for the previous example, clearly demonstrating the nonmonotonicity in λ and that standard continuation in λ would fail.

As a second example, consider the system given by

$$A = \begin{pmatrix} 0 & 1.0000 \\ -9.8696 & -0.0001 \end{pmatrix},$$

$$C = (10.0637 \quad -9.9363), \quad B = C^T$$

$$E_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad E_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad E_{1\infty} = E_1, \quad E_{2\infty} = E_2$$

$$D_1 = \begin{pmatrix} 0 & 0 \\ 0.0482 & 0 \end{pmatrix}, \quad D_2 = (0 \quad 1)$$

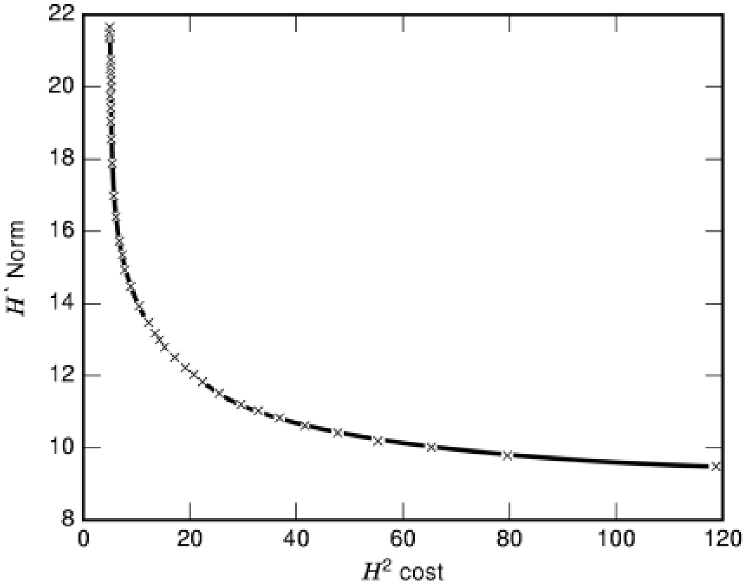

Fig. 4. $\|H(s)\|_\infty$ versus J for $n_c = 2$.

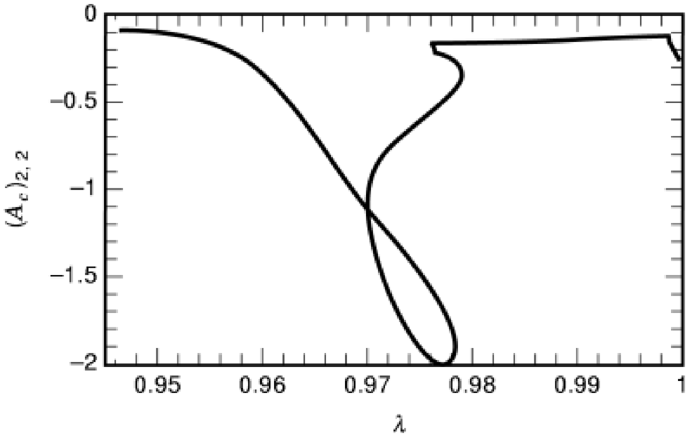

Fig. 5. $(A_c)_{2,2}$ versus λ for $\gamma = 3.8$ and $n_c = 2$.

The trade-off curve is shown in Fig. 6. The solutions of the auxiliary minimization problem are obtained for $\gamma \geq 0.032$. The ratio of $\|H(s)\|_\infty$ at $\gamma = 0.032$ to that at $\gamma = \infty$ is 0.69, which indicates that there is about 31% improvement in the H^∞ performance of the compensator with $\gamma = 0.032$ over the compensator without the H^∞ constraint.

The relative performance of homotopies based on the Ly parametrization, and overparametrization for the combined H^2/H^∞ model order reduction problem was reported in detail in (8,9,10,11). The Ly's formulation is very efficient but can fail to exist or lead to ill conditioning and it is conceivable that it will fail for some problems. This failure of existence in general is related to the insistence on using the minimal number of parameters $n_cm + n_cl$.

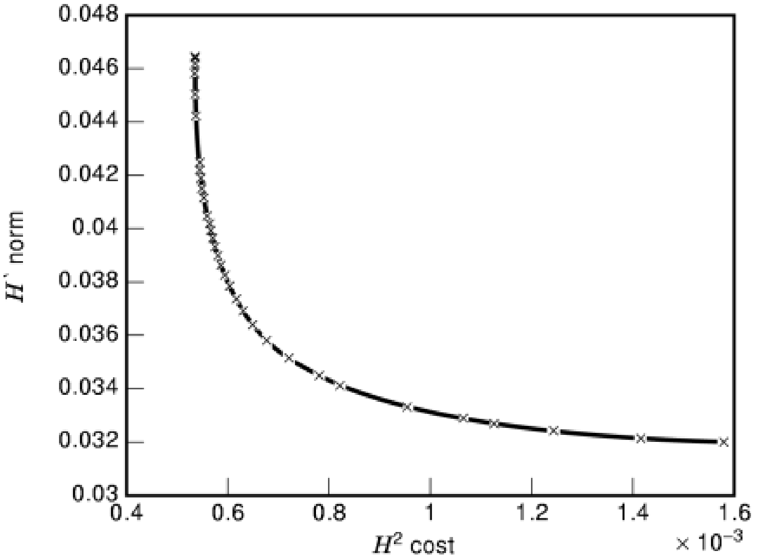

Fig. 6. $\|H(s)\|_\infty$ versus J for the second example.

By using $n_c(n_c + m + l)$ parameters, the over-parametrization formulation solves the ill-conditioning issue related to existence, but introduces a very high order singularity at the solution. It is doubtful whether the Hermite interpolation used here can handle a large problem with a singularity of order 100. A pragmatic suggestion is to try Ly's form first and then the over-parametrization form, switching if ill conditioning or failure occurs. The optimal projection equations formulation of (1) does not make structural assumptions (in fact is completely basis independent), but the optimal projection equations are very difficult and expensive to solve numerically. This cost can be reduced by exploiting tensor product structure and assuming monotonicity in λ of the homotopy zero curves, but Fig. 5 here shows that assumption is not tenable. The over-parameterization formulation makes no structural assumptions and is cheaper computationally than the optimal projection equations, but it is inherently singular at the solution with rank deficiency $n^{2}{}_{c}$, which will ultimately overwhelm the numerical linear algebra (8), (11).

The distributed code using PVM was run on a network of seven SGI Indigo^2 workstations. The data came from a control problem for suppressing vibrations in a string under transverse loading from a time varying disturbance force. For dimensions $n = 12$, 20, 28, 36 reduced order controllers of dimensions $n_c = 10$, 18, 26, 34 are sought, respectively.

The speedups versus the number of workstations are shown in Fig. 7 and Fig. 8 ($n = 36$, 28, 20, 12, top to bottom). Figure 7 shows the speedup versus the number of workstations when the master process and the slave processes are run on different machines, while Fig. 8 corresponds to the situation where the master process and a slave process with lower priority are run on one machine and the rest of the slaves are on other machines. For fair comparison all the speedups are computed relative to the results of the best optimized sequential code. In Fig. 7 two workstations correspond to the master process on one machine and the only slave on the other.

As shown in the figures, the speedup increases as the dimension of the problem increases, or as the number of workstations increases for a sufficiently large problem. The speedups from three scenarios (solid line—master and slaves on different machines; dash-dot line—master and a low priority slave on one machine and the rest of the slaves on others; dashed line—master and a slave with the same priority as the master on one machine) for $n = 20$ are plotted against the number of workstations in Fig. 9. If the number of workstations is < 4 it is better to use the second scenario. When the number of workstations is > 4, the speedup is higher if

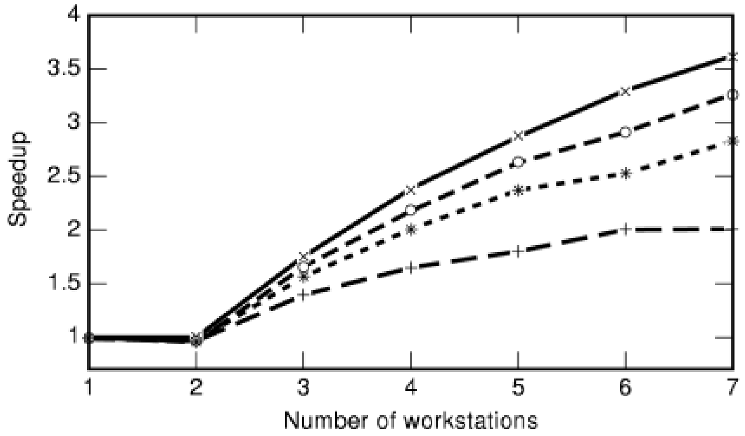

Fig. 7. Speedup with master and slaves on different machines.

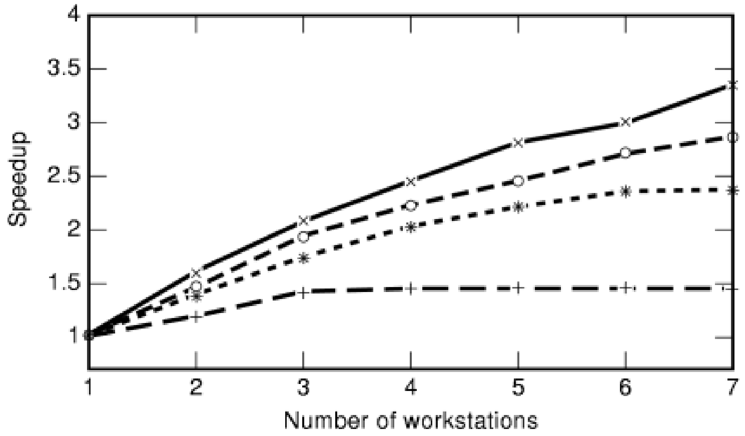

Fig. 8. Speedup when one slave is on the master machine.

all the processes including the master and the slaves are run on different machines. Similar results obtain for large n.

The same algorithm is implemented using the system function calls of an Intel Paragon and run on one with 28 processors at Virginia Polytechnic Institute and State University. Figure 10 shows the results obtained from the run for $n = 12$, 20, 28. The number of nodes varies up to 25. The highest curve corresponds to the speedup when $n = 28$ and the lowest corresponds to that when $n = 12$. As n increases, the advantage of parallel processing also increases. The highest speedup achieved for $n = 28$ using seven SGI Indigo^2 workstations is about 3.3 while the highest speedup using 25 nodes on an Intel Paragon is about 5.1. Comparing speedups is meaningful since the performance of a single SGI Indigo^2 processor is roughly comparable to that of a single i860XP Paragon node (actually, depending on the task, the 100 MHz R4000 Indigo^2 is faster by a factor of 2). However, the cost of the Intel Paragon is a factor of three times the cost of the SGI Unix workstation network. Much higher speedups are potentially possible with the Paragon, but not without considerable programming effort for this controller design problem.

The above methodology can be easily generalized to industrial design environments where software packages like MATLAB or *Mathematica* are often used. The sequential program for mixed-norm $H^2/^{H^\infty}$ LQG

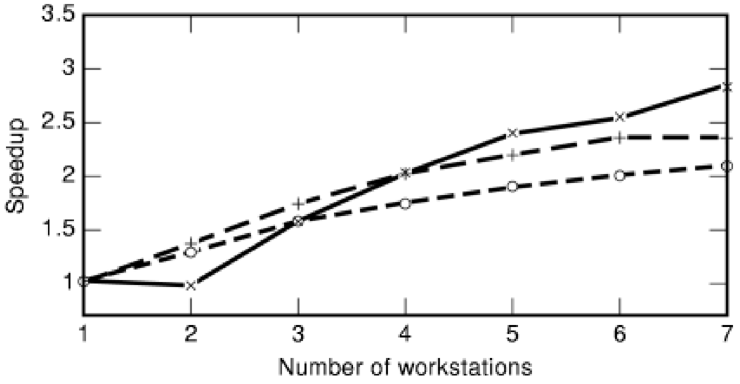

Fig. 9. Comparison of speedups.

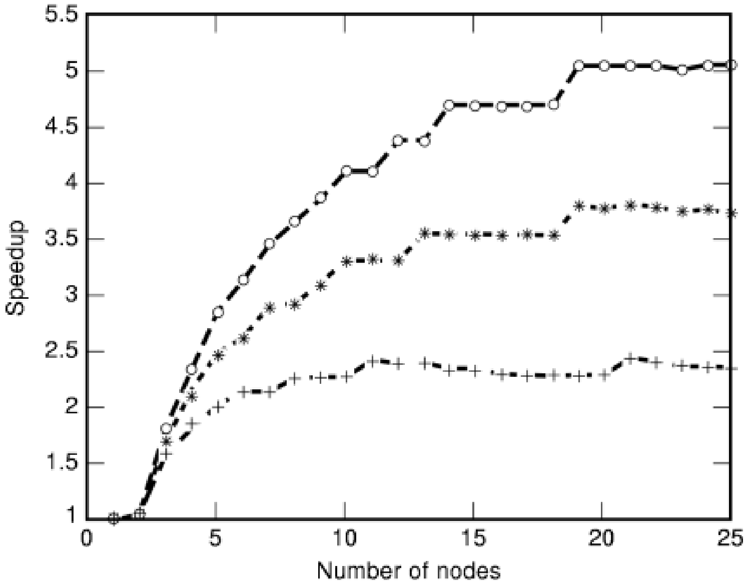

Fig. 10. Results for Intel Paragon XPE-28.

controller synthesis has been developed into a MATLAB package. It is easy to include this distributed implementation into the MATLAB package. Installation requires two steps: the first one is to install PVM on the network of workstations, and the second is to create a file in which all the worker machines on the network are listed (46). The execution of the distributed program from within an interactive design environment, for example, MATLAB, can be done by using a MATLAB function defined in a MATLAB .m file, in which Unix shell commands will start the PVM daemons if they have not already been started, and will then execute the distributed code.

Acknowledgment

The work of Yuzhen Ge and Emmanuel G. Collins, Jr. was supported in part by Air Force Office of Scientific Research grant F49620-95-1-0244 and the work of Layne T. Watson was supported in part by Air Force Office of Scientific Research grant F49620-92-J-0236 and Department of Energy grant DE-FG05-88ER25068/A004.

BIBLIOGRAPHY

1. D. S. Bernstein W. M. Haddad LQG control with an H_∞ performance bound: A Riccati equation approach, *IEEE Trans. Autom. Control.*, **AC-34**: 293–305, 1989.
2. W. M. Haddad D. S. Bernstein Generalized Riccati equations for the full- and reduced-order mixed-norm H_2/H_∞ standard problem, *Syst. Control Lett.*, **14**: 185–197, 1990.
3. K. Glover D. Mustafa Derivation of the maximum entropy H^∞ controller and a state space formula for its entropy, *Int. J. Control*, **50**: 899–916, 1989.
4. D. Mustafa Relations between maximum entropy/H^∞ control and combined H^∞/LQG control, *Syst. Control Lett.*, **12**: 193–203, 1989.
5. K. Zhou *et al.* Mixed H^2 and H^∞ control, *Proc. Amer. Control Conf.*, 1013–1017, 1990.
6. P. P. Khargonekar M. A. Rotea Mixed H^2/H^∞ control: A convex optimization approach, *IEEE Trans. Autom. Control*, **36**: 824–837, 1991.
7. D. Žigić *et al.* Homotopy approaches to the H_2 reduced order model problem, *J. Math. Syst., Estimation, Control*, **3**: 173–205, 1993.
8. Y. Ge Homotopy algorithms for the H^2 and the combined H^2/H^∞ model order reduction problems, *M. S. thesis*, Dept. Computer Sci., Virginia Polytechnic Inst. & State Univ., Blacksburg, VA, 1993.
9. Y. Ge *et al.* Probability-one homotopy algorithms for full and reduced order H^2/H^∞ controller synthesis, *Optimal Control Appl. Methods*, **17**: 187–208, 1996.
10. Y. Ge L. T. Watson E. G. Collins, Jr. A comparison of homotopies for alternative formulations of the L^2 optimal model order reduction problem, *J. Computational Appl. Math.*, **69**: 215–241, 1996.
11. Y. Ge *et al.* Globally convergent homotopy algorithms for the combined H^2/H^∞ model reduction problem, *J. Math. Syst., Estimation, Control*, **7**: 1997.
12. E. G. Collins, Jr. W. M. Haddad S. S. Ying Construction of low authority, nearly non-minimal LQG compensator for reduced-order control design, *preprint*, October, 1993.
13. L. T. Watson R. T. Haftka Modern homotopy methods in optimization, *Comput. Methods Appl. Mech. Eng.*, **74**: 289–305, 1989.
14. L. T. Watson S. C. Billups A. P. Morgan HOMPACK: A suite of codes for globally convergent homotopy algorithms, *ACM Trans. Math. Software*, **13**: 281–310, 1987.
15. M. Jamshidi An overview of the solutions of the algebraic matrix Riccati equation and related problems, *Large Scale Syst.*, **1**: 167–192, 1980.
16. M. A. Shayman Geometry of the algebraic Riccati equation, *SIAM J. Control Optim.*, **21**: 375–409, 1983.
17. I. Gohberg P. Lancaster L. Rodman On Hermitian solutions of the symmetric algebraic Riccati equation, *SIAM J. Control Optim.*, **24**: 1323–1334, 1986.
18. I. Gohberg P. Lancaster L. Rodman *Invariant Subspaces of Matrices with Applications*, New York: Wiley, 545, 1986.
19. V. Kučera A contribution to matrix quadratic equations, *IEEE Trans. Automat. Control*, **17**: 344–347, 1972.
20. P. Lancaster L. Rodman Existence and uniqueness theorems for the algebraic Riccati equation, *Int. J. Control*, **32**: 285–309, 1980.
21. C. Kenney A. Laub E. Jonckheere Positive and negative solutions of dual Riccati equations by matrix sign function iteration, *Syst. Control Lett.*, **13**: 109–116, 1989.
22. B. P. Molinari The time-invariant linear-quadratic optimal control problem, *Automatica*, **31**: 347–357, 1977.
23. J. E. Potter Matrix quadratic solutions, *SIAM J. Appl. Math.*, **14**: 496–501, 1966.
24. D. L. Kleinman On an iterative technique for Riccati equation computations, *IEEE Trans. Autom. Control*, **AC-13**: 114–115, 1968.

25. A. J. Laub A Schur method for solving algebraic Riccati equations, *IEEE Trans. Automat. Control*, **AC-24**: 913–921, 1979.
26. A. J. Laub Numerical linear algebra aspects of control design computations, *IEEE Trans. Automat. Control*, **AC-30**: 97–108, 1985.
27. A. J. Laub Invariant subspace methods for the numerical solutions of Riccati equations, in S. Bittanti, A. J. Laub, and J. C. Willems (eds.), *The Riccati Equations*, New York: Springer-Verlag, 163–195, 1991.
28. P. Van Dooren A generalized eigenvalue approach for solving Riccati equations, *SIAM J. Sci. Stat. Comp.*, **2**: 121–135, 1981.
29. W. F. Arnold A. J. Laub Generalized eigenproblem algorithms and software for algebraic Riccati equations, *Proc. IEEE*, **72**: 1746–1754, 1984.
30. T. Pappas A. J. Laub N. R. Sandell, Jr. On the numerical solution of the discrete-time algebraic Riccati equation, *IEEE Trans. Automat. Control*, **AC-25**: 631–641, 1980.
31. S. S. L. Chang T. K. C. Peng Adaptive guaranteed cost control of systems with uncertain parameters, *IEEE Trans. Automat. Control*, **AC-17**: 474–483, 1972.
32. T. Kailath Some Chandrasekhar-type algorithms for quadratic regulators, *Proc. IEEE Conf. Decis. Control*, New Orleans, LA, 219–223,1972.
33. K. Ito R. K. Powers Chandrasekhar equations for infinite dimensional systems, *SIAM J. Control Optim.*, **25**: 596–611, 1987.
34. R. Byers A Hamiltonian QR algorithm, *SIAM J. Sci. Stat. Comp.*, **7**(1): 212–229, 1986.
35. M. Hazewinkel R. E. Kalman On invariants, canonical forms and moduli for linear, constant, finite dimensional, dynamical systems, *Proc. Math. Syst. Theory*, Udine, Italy, 1975.
36. K. Glover J. C. Willems Parameterization of linear dynamical systems: Canonical forms and identifiability, *IEEE Trans. Autom. Control*, **AC-19**: 640–651, 1974.
37. A. C. Antoulas On canonical forms for linear constant systems, *Int. J. Control*, **33**: 95–122, 1981.
38. G. O. Correa K. Glover Pseudo-canonical forms, identifiable parameterizations and simple parameter estimation for linear multivariable systems: Input-output models, *Automatica*, **20**: 429–442, 1984.
39. P. K. Kabamba Balanced forms: Canonicity and parametrization, *IEEE Trans. Autom. Control*, **30**: 1106–1109, 1985.
40. R. Ober Balanced parametrization of classes of linear systems, *SIAM J. Control Optim.*, **29**: 1251–1287, 1988.
41. L. D. Davis E. G. Collins, Jr. S. A. Hodel A parametrization of minimal plants, *Proc. 1992 Amer. Control Conf.*, Chicago, IL, 355–356, June 1992.
42. U.-L. Ly A. E. Bryson R. H. Cannon Design of low-order compensators using parameter optimization, *Automatica*, **21**: 315–318, 1985.
43. A. Yousuff R. E. Skelton A note on balanced controller reduction, *IEEE Trans. Autom. Control*, **AC-29**: 254–257, 1984.
44. R. H. Bartels G. W. Stewart Solution of matrix equation $AX + XB = C$, *Comm. ACM*, **15**: 820–826, 1972.
45. R. H. Cannon, Jr. D. E. Rosenthal Experiments in control of flexible structures with noncolocated sensors and actuators, *AIAA J. Guid. Control Dynam.*, **7**: 546–553, 1984.
46. A. Geist *et al.* PVM 3 User's Guide and Reference Manual, *Oak Ridge National Laboratory*, ORNL/TM-12187, Oak Ridge, TN, 1993.

YUZHEN GE
Butler University
LAYNE T. WATSON
Virginia Polytechnic Institute and State University
DENNIS S. BERNSTEIN
University of Michigan
EMMANUEL G. COLLINS, JR.
Florida A&M/Florida State University