ROBOT PATH PLANNING

PLANNING AS A REACTION TO ANTICIPATION

Historical Overview

Path planning by a robot seems to be a natural step preceding the functioning of its motion control system. Actually, path planning is just a component of the more general paradigm of *motion planning,* which is important not only for robots but also for other objects and systems. Since the 1960s, interest in motion planning has grown and spread to various domains of application.

Motion planning is an intersection of three weakly related scientific paradigms: operation research (OR), artificial intelligence (AI), and control theory. OR emerged in the 1940s and spurred the analysis of queues, graph theory, and methods of optimization. As an AI extension in the 1960s, the study of planning targeted corresponding processes of human cognition, and the first effort in explicit analysis of planning algorithms was related to human thought simulation (1). Newell, Simon, Nilsson, and other prominent researchers in AI developed the fundamentals for the existing results in the area of robot motion planning. Traditionally for AI, planning was not involved into any "dynamics," which was always considered the domain of control theory.

In the 1970s, Fu, Saridis, and their students initiated research of control systems, which incorporated planning and recognition (2,3), and which eventually brought to fruition a new direction: intelligent control (4). As a discipline, intelligent control blends OR, AI, and control theory. It is concerned with analysis of planning, particularly for robotics. After this, the mainstream specialists in control theory realized that the so-called "reference trajectory," which is always regarded as the input to control systems, should be considered a "plan" and computed as a part of the design process. However, the traditional control specialists considered extraneous everything not immersed into a rigid mathematical paradigm.

STRIPS (5,6) and A* (7) became classical fundamentals of planning in robotics. The subsequent development in the area of robot path planning branched enormously:

- The problems of representation turned out to be very critical.
- It became clear that both combinatorics of tasks and dynamics of systems are intertwined.
- Planning grew up hierarchically.
- The complexity of computations became the real limitation for the development of theories.

The milestones in the evolution in the area of motion and path planning are as follows:

- In 1966, Doran and Michie applied a graph-theoretic mechanism for path planning (8).
- In 1968, Howden introduced the "Sofa Problem," treating the geometric problem of motion planning (9).
- In 1968, the A* algorithm was introduce by Hart, Nilsson, and Rafael (7).
- In 1971, STRIPS was presented by Fikes, Hart, and Nilsson (5,6).
- In 1979, the concept of search was attempted for dealing with obstacles by Lozano-Perez and Wesley (10).
- In 1979, Albus introduced the methodology of task decomposition for hierarchical systems; later it became a part of the NIST-RCS methodology, with nested planning processes at all levels of the control hierarchy (11).
- In 1981, Lozano-Perez applied "configuration space" to manipulator's planning (12).
- In 1983, Julliere, Marce, and Place outlined their mobile robot with planning via tessellated space (13).
- In 1984, Chavez and Meystel (14) introduced a concept of searching in the space of various (nonuniform) traversability.
- In 1985, Hoperoft, Joseph, and Whitesides analyzed the geometry of robotic arm movement in 2-D bounded regions (15).
- In 1986, Meystel demonstrated that the most efficient (least computational complexity) functioning of a multilevel learning/control systems with search for the planning can be provided by a proper choice of a ratio of lower level/higher level of resolution (16). This concept of planning/control hierarchy became a strong theoretical support for the hierarchical architecture of intelligent system.
- In 1985 to 1987, Arbib's school of control via "schemata" came up with a numerous schemes of "reactive" behavior. This gave birth to a multiplicity of robot control concepts which explored and exercised reactive behavior generation.
- During the period from 1985 to 1995, many researchers associated problems of robotic motion planning with short-term (local) reactive behavior (e.g., "obstacle avoid-

ance"). Nevertheless, the interest to the search in the state space was perpetuating.

- In the meantime, the primary focus of robotics shifts to the area of systems that do not require any planning (robotics with "situated behavior"). Thus the interest in planning diminishes (Brooks, MIT, Arkin, Georgia Tech) and the curiosity of researchers shifts toward emerging phenomena in robots with rudimentary intelligence.
- In 1991, a comprehensive text has been published by Latombe (17), which outlines most of the theories and experiences approved by the practice in a variety of applications. It happened a whole decade after the first textbook edited by Brady, J. Hollerbach, Johnson, Lozano-Perez, and Mason (18).

Ten years of research and experience (1982-1991) helped to clarify the important maxim: the process of robot motion planning can be performed efficiently only by searching within the state space and thus, determining both the final goal, and the trajectory of motion leading to this goal. At the present time, search in the state space is a prevailing general technique broadly applied for the algorithms of planning. Nevertheless, many other concepts and systems exist too, in a multiplicity of research schools and domains of application.

Definitions Related to Planning

The following definitions are typical for the common dictionaries (e.g., Merriam-Webster's *Collegiate Dictionary*)

Plan (as a noun) 1: a drawing, or diagram drawn on a plane: as (a) a top or horizontal view of an object, (b) a large-scale map of a small area; 2(a) a method for achieving an end, (b) an often customary method of doing something; (c) a procedure: a detailed formulation of a program of action; (d) goal, aim; 3: an orderly arrangement of parts of an overall design or objective; 4: a detailed program (as for payment or the provision of some service).

Plan (as a verb) 1: to arrange the parts of: design; 2: to devise or project the realization or achievement of (a program); 3: to have in mind: intend: to make plans.

These definitions can be applied to a module of intelligent system which receives a goal, retrieves relevant knowledge in the world model, and creates strings of tasks for the actuators (or the similar modules below in the hierarchy; the latter consider them their "goals").

The professional definitions for specialists involved in planning and control of robots, are recommended by the NIST research report on Behavior Generation in Intelligent Systems (including robots) (19). The system of behavior generation is supposed to be constructed out of BG-modules, each module equipped with a planner. Within this paradigm, *plan* is the set of schedules for the group of agents which are supposed to execute these schedules as a cooperative effort and accomplish the required job (achieve the goal) as a result of this effort. To find this set of schedules different combinations of agents should be tested, and different schedules should be explored. Plan is also defined:

As the course of events determined within BG-module which is supposed to be reproduced in the world to achieve the goal in the desirable fashion

Or—as a description of the set of behaviors which lead to the goal in the desirable fashion. This description is represented as a set of "schedules"

Or—it is a state space trajectory that describes the behavior of system leading to the goal and providing satisfaction of constraints and conditions on some cost-function, or cost-functional (these conditions might include: having the value of this cost-function/cost-functional within some interval, maximizing, or minimizing it)

Thus, *plan* controls the system. It consists of two major components: the final state, which should be achieved in the end of the planning interval, and the string of the intermediate states, which are often supplemented by their time-schedule and are bounded in the value of admissible error.

Plan consists of task space/time decompositions, such as the subtasks that are distributed in space and time. It may be represented as a PERT chart, Gannt diagram, a state transition graph, a set of schedules complemented by the account of resources required (e.g., bill of materials, tools and manpower requirements, delivery schedules, and cost estimates). Each plan is characterized by its *goal, time horizon, set of agents (performers),* and its *envelope.*

Plan, Optimal is the plan that leads to the goal achievement while minimizing (or maximizing) a particular cost-function, or a cost-functional. Optimal plans can be found (synthesized) only as a result of the comparison among all alternatives of feasible (admissible) plans.

Plan, Satisficing one of the admissible plans which is within a narrowed set of constraints. It is one of the state space trajectories which is constructed within the desirable boundaries specified by a customer who does not want to determine the cost-function. In other words, this is a sufficient, satisfactory, but not necessarily "the best" plan.

Plan, Spatial is the state space trajectory (in the enhanced state space which includes inputs, outputs, and states of the system.) The state space trajectory should be represented at the output of the planning submodule as the result of selection of agents and jobs assigned to them, their responsibilities, and criteria of their performance.

Plan, Temporal is explained below, under schedule.

Plans, Admissible are all meaningful plans that can be built within the specified constraints.

Planning is the design of the course of events determined within BG-module; design of the desirable state space trajectory; design of the feedforward control function, and, thus, the future for the system. Planning is performed in an assumption that we know the agents of the adjacent higher level of resolution which will cooperate in the process of the further delineation of the plan. This assumption corresponds to one particular alternative of the solution. Another alternative has another assumption about performing agents and leads to another plan. The design of the desirable motion of the system entails that many supportive components of operation also should be planned: the algorithms of feedback compensation, inputs to the energy converters, the scope of sensing (focus of attention), and others.

Planning envelope is a subset of the state space with a corresponding world model which is submitted to BG at the higher level of resolution for refinement. Upon completion of the planning process at a level, a part of this plan should be refined by searching for a more precise solution in the limited envelope around the planned trajectory. A subset of the plan (for a limited time smaller than the planning horizon of the level under consideration) is submitted to BG unit of the higher level of resolution for a refinement.

Planning horizon is the time interval within which a plan is meaningful. The degree of belief for each future state of the plan falls off as time t grows large because the stochastic component of the operation affects the verifiability of the results. For some particular value of time in the future the degree of belief is lower than the degree required for the decision making process. This value of time is called "planning horizon."

Planning strategy orientation toward receiving either the optimal or the satisficing plan.

Replanning is the process of planning which is performed if the top-down and bottom-up processes of plan propagation did not converge. The need in replanning can emerge (a) if the initially selected version of plan distribution failed, (b) if the prescribed conditions of compensation fail to keep the process within the prescribed boundaries, (c) if the world model has changed or (d) if the goal has changed.

Resources the following resources are usually taken into account for constructing the cost-function: time, energy, materials, remaining life-span of the system, the degree of fault-tolerance, and money.

Resolution is the property of the level of hierarchy which limits the distinguishability of details. (Synonyms are *scale, granulation, coarseness*).

Schedule is another term for the "temporal plan"; it is the description of the development of the process in time. It obtained by computing the state space trajectory within the time domain. The schedule should focus upon the start and the end events and provide for coordination, reduced queues, and elimination of the "bottlenecks." Schedule can be also defined as a job-time event-gram.

Scheduling is the process of outlining the temporal development of the motion trajectory.

The following supporting definitions might be useful for interpreting the other sections of this article.

Supporting Definitions

- *Behavior* is the ordered set of consecutive-concurrent changes (in time) of the states registered at the output of a system (in space). In a goal-oriented system, behavior is the result of executing a series of tasks.
- A *task* is a piece of work to be done, or an activity to be performed. It can be described as a data structure representing the assignment.
- *Action* is an effort generated by the actuator producing changes in the world.
- *Space-time* (spatio-temporal) representation presuming description of the process as a sequence of time-tagged states (temporal sequence) in which each state is a vector in the space with coordinates corresponding to all variables of the process (including input, output, and inner states variables).
- *Goal* is the state to be achieved or an objective toward which task activity is directed (e.g., a particular event). A goal can be considered an event which successfully terminates the task.
- A *task command* is an instruction to perform a named task. This is an assignment presented in the code pertaining to a particular module of the system. A task command may have the form:

DO ⟨Task_name (parameters)⟩ AFTER ⟨Start State (or Event)⟩
UNTIL ⟨Goal State (or Event)⟩

Summarizing the specific definitions above, the following definition of planning can be considered instrumental:

- *Planning* is a process of searching for appropriate future trajectories of motion leading to the goal. Searching is performed within the system of representation.

Planning as a Stage of Control

A general control diagram is shown in Fig. 1. It starts with a reference trajectory ("desired motion trajectory") at the input, and it ends with the output motion. The measured output differs from the desired motion, and the difference between them enables the feedback controller to perform the compensation. It turns out that the "feedforward" part of the control systems plays partially a role of a planner.

Intuitively, it is to be expected that reacting to error will be a relatively slow process compared to the predictive correction that is available via feedforward channel. By the virtue of existing outside the immediate scope of the feedback loop, the feedforward controller injects a priori known bias into the operation of that loop. Regardless of whether this bias consists of a nominal command applied by an expert to affect the continuous operation of a machine in a factory, or whether its is a linearizing or decoupling torque generation scheme for a robotic manipulator, the planned command input is produced on the basis of the analysis of a system model in some form, mathematical, linguistic (or both) in order to improve the performance of the overall system.

If a reference trajectory for such a system has been synthesized, then it is necessary for the feedforward controller to be the inverse dynamical model of the plant. Thus the only task for the feedback part is to cancel the effects of *unmodeled* dynamics and disturbances. The transfer function can be simplified to unity if the plant and feedforward controller are exact inverses of each other. In fact, the bulk of the existing results in the area are based on this assumption. The work of Brockett (20) is generally acknowledged as the first formal treatment of the problem of the inversion of multivariable linear time invariant systems and numerous papers (21–24) list extensions of those results to nonlinear, time-varying, and discrete-time cases.

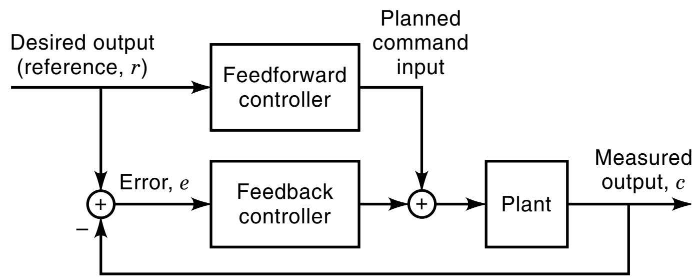

Figure 1. Combined feedforward/feedback control architecture.

Thus, if the reference trajectory is known, then planning boils down to inverting this known trajectory. The general approach to planning as a synthesis of feedforward commands is to implement an algorithmic procedure ("approximate inversion") for the determination of a nominal input function or trajectory which leads to approximate tracking of the reference trajectory.

Approximate inversion is an algorithmic procedure for planning an input function in some admissible input set which, when applied to a plant, causes its outputs to follow a trajectory which minimizes the deviation of those outputs from a prescribed trajectory over some closed interval of time. An input so determined shall be referred to as an *approximate inverse* of the reference trajectory over the input set.

Since the reference trajectory should be found, the full planning process should involve finding it, for example, by optimizing the output motion of the system. Specifically, the optimal control of systems via the calculus of variations and Pontryagin's principle should, in theory, provide both the reference trajectories for a system as well as the inputs required to generate them. In practice, optimal solutions are hard to generate for all but the simplest problems. In many other cases, such as those involving systems with large, distributed parameters, and computer-based models, it is not possible to apply the classical theory at all.

Optimization is typically performed by using the tools of searching. Search should be performed within some envelope around the desired trajectory. Thus an envelope around the desired trajectory should be submitted to the input as a primary assignment. This envelope contains the initial and the goal-points. This envelope encloses the space for the subsequent search of the optimum trajectory.

PLANNING AS A PART OF BEHAVIOR GENERATION

Behavior Generation

Robotics is the integrated domain providing for blending the goals and testing the means of achieving them, that is, a domain with a direct need for planning. In 1983, T. Lozano-Perez introduced the idea of search in a "configuration space." From the experience of using this search, it became clear that the exhaustive search would be computationally prohibitive if the configuration space is tessellated with the accuracy required for motion control. But this theory made one important thing obvious: planning is an apogee of synthesizing the admissible alternatives and searching for the trajectories entailed by these alternatives.

This development helped to realize that planning should combine the exhaustive (or meaningfully thorough) search off-line, as a part of the algorithm of an off-line control. It was about at this time that engineers stopped talking about control of actions and introduced a more balanced term-behavior generation. The latter became a code word for the joint process of testing the alternatives within the mechanism of "planning" (open loop, feedforward control) blended with the on-line finding the alternatives of feedback for error compensation (closed-loop control, or "execution").

Behavior generation alludes to many mechanisms of planning and execution. At the present time, these mechanisms cannot be considered fully understood, and the general theory of planning can hardly be immediately attempted. There is merit in discussing a subset of problems in which the goal is determined as attainment of a particular state.

Behavior generation (19) can have many different mechanisms of planning and execution. These mechanisms are not well known. A subset of problems will be discussed, in which the goal is defined as attainment of a particular state. Most of the realistic problems can be translated in this paradigm. Other types of problems can also be imagined: in chess, the goal is clear (to win) but this goal cannot be achieved by simply reaching a particular position in a space (even in a descriptive space). Most of the problems related to the theory of games and linked with pursuit and evasion are characterized by a similar predicament and are not discussed here.

Problems Related to Planning

Any problem of planning is associated with

- Actual existence of the present state (PS)
- Actual, or potential existence of the goal state (GS)
- Knowledge of the values for all or part of the states as far as some particular goal is concerned (KS)

The cumulative costs of trajectories to a particular goal (or goals) can be deduced from this knowledge. On the other hand, the knowledge of costs for the many trajectories traversed in the past can be obtained, which is equivalent to knowing cumulative costs from the initial state (PS) to the goal state (GS) (from which the values of the states can be deduced).

In other words, any problem of planning contains two components: the first one is to refine the goal (i.e., bring it to the higher resolution.) The second one is to determine the motion trajectory to this refined goal. These two parts can be performed together, or separately. Frequently they are dealt with separately. In the latter case they are formulated as follows:

- Given PS, GS, and KS (all paths), find the subset of KS with a minimum cost, or with a preassigned cost, or with a cost in a particular interval.
- Given PS and GS from the lower resolution level and KS (all paths), find the GS with a particular value.

In (25,26) two important issues are introduced for the area of planning: controllability and recognizability. The controllability issue arises when the number of controls is smaller than the number of independent parameters defining the robot's configuration. The recognizability issue occurs when there are errors in control and sensing: how well the robot can recognize goal achievement. Both issues can affect the computational complexity of motion planning. The set of controllability, recognizability, and complexity is especially important to the development of autonomous robots.

Planning in a Representation Space with a Given Goal

The world is assumed to be judged upon by using its state space (or the space of representation), which is interpreted as a vector space with a number of important properties. Any activity (motion) in the world (space of representation) can be characterized by a trajectory of motion along which the "working point" or "present state" (PS) is traversing this space from one point (initial, or state, IS) to one or many other states (goal states, GS). The goal states are given initially from the external source as a "goal region," or a "goal subspace," in which the goal state is not completely defined in a general case.

From the point of view of planning, state space does not differ from the configuration space. Indeed, the upcoming behavior is represented as a trajectory in the state-space (and/or configuration space). One of the stages of planning (often the initial one) is defining where exactly is the GS within the "goal region" (which was the "goal state" at the lower resolution). In many practical problems, the designer should focus upon planning procedures in which one or many GS remain unchanged through the entire period of their functioning (before they are achieved). Traversing from IS to GS is associated with consuming time, or another commodity (cost). So, the straightforward exhaustive search is feasible which allows for exploring all possible alternatives.

Planning as a Reaction to Anticipated Future

Researchers in the area of reactive behavior introduced a method of potential fields for producing comparatively sophisticated obstacle avoiding schemes of motion. Reactive behavior is considered to be the antithesis of planning. This is not so. Motion based upon planning can be called reactive, too. The difference is that in reactive behavior robots usually react to the present situation. In the system with planning, one reacts, too: but one reacts to the anticipated future.

Thus planning can be considered an anticipatory reactive behavior. The difference is in the fact that anticipation requires representation richer than the simple reactive behavior requires.

Types of Representation Available for Planning

All representation spaces are acquired from the external reality by learning processes. Many types of learning are mentioned in the literature (supervised, unsupervised, reinforcement, dynamic, PAC, etc.). Before classifying a need in a particular method of learning and deciding how to learn, one must figure out exactly what is to be learned. It is important to find out whether the process of learning can be separated into two different learning processes:

1. That of finding the world representation
2. That of finding the appropriate rules of action

or these two kinds of learning are just two interrelated sides of the same core learning process.

The following knowledge should be contained in the representation space. If no GS is given, any pair of state representations should contain implicitly the rule of moving from one state to another. In this case, while learning, one inadvertently considers any second state as a provisional GS.

Call "proper" representation a representation similar to the mathematical function and/or field description: at any point the derivative is available together with the value of the function; the derivative can be considered an action required to produce the change in the value of the function.

Call "goal-oriented" a representation in which the value of the action at each given point is required for describing not the best way of achieving an adjacent point, but the best way of achieving the final goal. Both "proper" and "goal-oriented" representation can be transformed in each other.

Artifacts of Representation Space

Representation of the world can be characterized by the following artifacts:

- Existence of states with boundaries determined by the resolution of (each state is presented as a tessellatum, or an elementary unit of representation, the lowest possible bounds of attention)
- Characteristics of the tessellatum, which is defined as an indistinguishability zone (consider that resolution of the space shows how far the "adjacent" tessellata, or states are located from the "present state" (PS) tessellatum)
- Lists of coordinate values at a particular tessellatum in space and time
- Lists of actions to be applied at a particular tessellatum in space and time in order to achieve a selected adjacent tessellatum in space and time
- Existence of strings of states intermingled with the strings of actions required to receive next consecutive tessellata of these strings of states
- Boundaries (largest possible bounds of the space) and obstacles
- Costs of traversing from a state to a state and through strings of states

In many cases, the states contain information which pertains to the part of the world which is beyond the ability to achieve it, and this part is called "environment." Another part of the world is to be controlled: this is the system for which the planning is to be performed. It will be referred to frequently as "self." Thus part of the representation is related to "self," including knowledge about actions which this "self" should undertake in order to traverse the environment.

It is seen from the list of artifacts that all knowledge is represented at a particular resolution. Thus the same reality can be represented at many resolutions and the "multiresolutional representation" is presumed. The system of representation is expected to be organized in a multiresolutional fashion. This will invoke the need to apply a number of special constraints and rules. The rules of inclusion (aggregation/decomposition) are especially important.

CLASSIFICATION OF ROBOT PLANNING PROBLEMS

Geometric Models

This domain is strongly linked with practical problems. It also generates a variety of famous theoretical problems: the "sofa" problem evolved into "piano-movers" problem. A thorough survey is given in (27). An interesting geometric model based upon Snell's law is presented in (32).

Collision-Free Robot Path

Most of the FINDPATH algorithms of the 1980s are based upon searching for a minimum path string of vertices within the socalled "visibility" graph [a graph comprising all vertices of the polygonal objects connected with visibility lines (28–31)].

Nonholonomic Path Planning

Mobile robots can be considered single-body devices (car-like robots) or composed of several bodies (tractors towing several trailers sequentially hooked). These robots are known to be nonholonomic, that is, they are subject to nonintegrable equality kinematic constraints involving the velocity. The number of controls is smaller than the dimension of the configuration space. The range of possible controls has additional inequality constraints due to mechanical stops in the steering mechanism of the tractor. It is demonstrated for the nonholonomic multibody robots that the Controllability Rank Condition Theorem is applicable even when there are inequality constraints on the velocity, in addition to the equality constraints (33–34).

Planning in Unknown, or Partially Known Environment

Planning in unknown environment is a problem that defies orientation to derive the search process from the concrete knowledge of the environment. Indeed, the map of a maze might be unknown but the strategy of behavior in a maze should exist. In this and numerous other situations it is required to have a "winning" strategy of actions under condition of lacking or absent information. There is an area of research oriented toward finding the most general rules of dealing with different types of environment (56–58).

Planning in Redundant Systems

Nonredundant systems have a unique trajectory of motion from a state to a state. Redundant system is defined as a system in which more than one trajectory of motion is available from one state to another. It can be demonstrated for many realistic couples of "system-environment" that

- They have a multiplicity of traversing trajectories from a IS to a GS.
- These trajectories can have different costs.

These systems contain a multiplicity of alternatives of space traversal. Redundancy grows when the system is considered to be a stochastic one. The number of available alternatives grows even higher when one considers also a multiplicity of goal tessellata of a particular level of resolution under the condition of assigning the goal at a lower resolution level, which is the fact in multiresolutional systems (e.g., NIST-RCS).

In nonredundant systems there is no problem of planning: only one trajectory of motion is available. Since the trajectory of motion to be executed is a unique one, the problem is to determine this trajectory and to provide tracking of it by an appropriate control system. Many research results demonstrate that redundancy can be considered an important precondition (a) for the need of planning, and (b) for performing planning successfully (59–62).

Figure 2 is a demonstration of the realistic situation of redundancy typical for the most of the planning problems: there are many paths from two geographical points in the relief demonstrated in the picture. If the only requirement is "minimum-time," a comparison of several paths will decide the path-winner. However, by introducing additional preferences and components of the cost-functional, the redundancy can be effectively reduced and even eliminated.

Uncertainty and Probabilistic Techniques for Path Planning

Most of the techniques for searching the minimum-cost paths on the graph are deterministic ones, and introduction of uncertainty became a new source of challenge (35–37). An approach to motion planning with uncertainty for mobile robots is introduced in (38). Given a model of the robot's environment, a "sensory uncertainty field" (SUF) is computed over the robot's configuration space. At every configuration, the SUF is an estimate of the distribution of possible errors in the "sensed configuration" and it is computed by matching the data given by the robot sensors against the model. A planner is using SUF to generate paths minimizing the expected errors. SUF has been explored for a classical line-striping camera/laser range sensor.

Planning relies on information that becomes available to the sensors during execution, to allow the robot to correctly identify the states it traverses. The set of states should be chosen, the motion command should be associated with every state, and the state evolution should be evaluated. The interdependence of these tasks can be avoided by assuming the existence of landmark regions in the workspace, which could be considered "islands of perfection," where the position sensing and motion control are accurate (39).

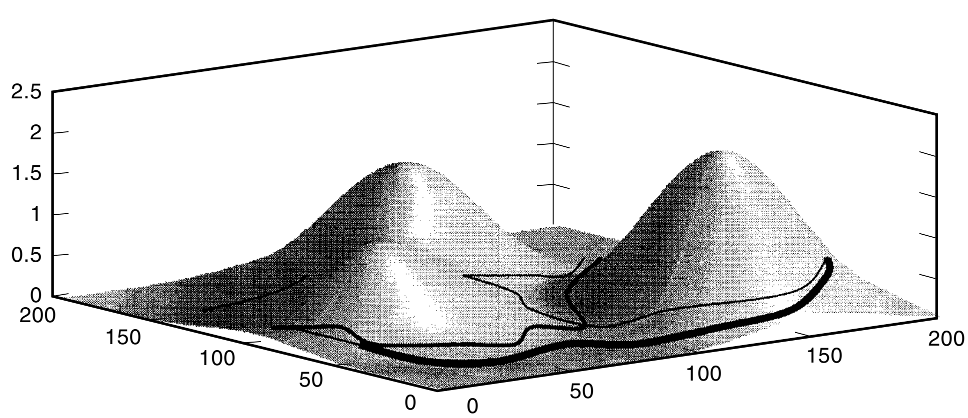

Figure 2. Multiplicity of plan alternatives.

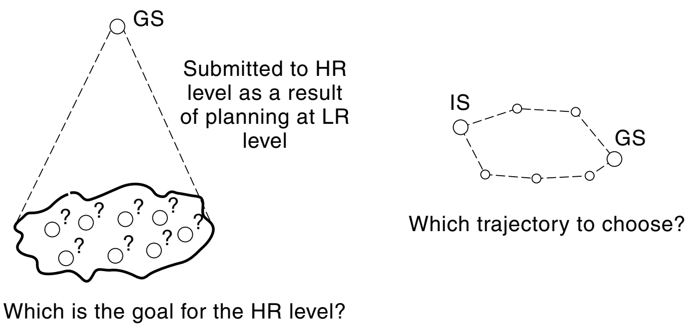

Figure 3. Two parts of planning problem.

PLANNING OF ACTIONS VERSUS PLANNING OF STATES

Algorithms of Planning

Planning constructs the goal states, and/or the preferable strings of states connecting the present state with the goal states. One of the successful techniques is associated with task decomposition (40). Task decomposition is related to the consecutive refinement, that is, to consecutive increase of the resolution of representation for both actions and states.

The first component of the planning algorithm is translation of the goal state description from the language of low resolution to the level of high resolution. Frequently, it is associated with increasing of the total number of the state variables. In all cases, it is associated with increasing the scale of representation, or with reduction of the indistinguishability zone, or the size of the tessellatum associated with a particular variable (see Fig. 3).

The second component is the simulation of all available alternatives of the motion from the initial state (IS) to one or several goal states (GS) and selection of the "best" trajectory. Procedurally, this simulation is performed as a search, that is, via combinatorial construction of all possible strings (groups). To make this combinatorial search for a desirable group more efficient one reduces the space of searching by focusing attention, that is, by preselection of the subset of the state space for further searching.

Thus all planning algorithms consist of two components: (1) a module for exploration of spatial distribution of the trajectory, and (2) a module for exploration of the temporal distribution. No algorithm of planning is conceivable without these two components.

The need in planning is determined by the multialternative character of the reality. The process of planning can be made more efficient by using appropriate heuristics.

Visibility-Based Planning

The "intelligent observer" (IO) is introduced in (41) as a mobile robot that moves through an indoor environment while autonomously observing moving targets selected by a human operator. The robot carries one or more cameras, which allow it to track objects while at the same time sensing its own location. It interacts with a human user, who issues task-level commands, such as indicating a target to track by clicking in a camera image. The user could be located far away from the observer itself, communicating with the robot over a network. As the IO performs its tasks, the system provides real-time visual feedback to the user. A prototype of the IO, which integrates basic versions of four major components: localization, target tracking, motion planning, and robot control, has been implemented. Initial experiments using this prototype, which demonstrate the successful integration of these components and the utility of the overall system, have been performed.

A particular problem of computing robot motion strategies is outlined in (42). The task is to maintain visibility of a moving target in a cluttered workspace. Both motion constraints (as considered in standard motion planning) and visibility constraints (as considered in visual tracking) are taken in account. A minimum path criterion is applied. Predictability of the target is taken in account. For the predictable case, an algorithm that computes optimal, numerical solutions has been developed. For the more challenging case of a partially predictable target, two on-line algorithms have been developed, each which attempts to maintain future visibility with limited prediction. One strategy maximizes the probability that the target will remain in view in a subsequent time step, and the other maximizes the minimum time in which the target could escape the visibility region.

Local Planning: Potential Field for World Representation

Genetic Search

The most pervasive method for navigating with minimal planning effort is using potential field construction around the obstacles (43,44). Potential field presumes adding to the world representation such properties that will increase the cost of moving into particular directions. An approach to robot path planning is proposed in (45), consisting of building and searching a graph connecting the local minima of a potential function defined over the robot's configuration space. The planner based on this approach allows to solve problems for robots with many more degrees of freedom. The power of the planner derives both from the "good" properties of the potential function and from the efficiency of the techniques used to escape the local minima of this function. The most powerful of these techniques is a Monte-Carlo technique, which escapes local minima by executing Brownian motions. The overall approach is made possible by the systematic use of distributed representations (bitmaps) for both the robot's workspace and configuration space. Genetic search is one of the tools for local planning. In some environments it gives positive results and can be recommended for use (46,47).

Global Planning: Search for the Trajectories

The most general way of planning is by global searching. It consists of the following stages:

1. Populate the world with the randomly assigned "points" that become vertices of the search graph.
2. Connect them in the vicinity.
3. Determine the cost of edges.
4. Run the graph search algorithm (e.g., Dijkstra algorithm or A*).

There are some problems that can be resolved in each particular case. Indeed, the "density" of future vertices of the search graph is to be selected. The concept of "vicinity" should be discussed, and the value of this vicinity should be properly evaluated. Different techniques of pruning the search-tree should be discussed. This area is explored in (48–52).

Several randomized path planners have been proposed (48–54). They are recommended to a variety of robots. A general planning scheme is introduced that consists of randomly sampling the robot's configuration space. The choice of points candidates can be determined by a relation between the probability of failure and the running time. The running time only grows as the absolute value of the logarithm of the probability of failure that one is willing to tolerate.

LINKAGE BETWEEN PLANNING AND LEARNING

Learning as a Source of Representation

Learning is defined as knowledge acquisition via experience of functioning. Thus learning is development and enhancement of the representation space under various goals. The representation can be characterized in the following ways:

- By a set of trajectories (to one or more goals) previously traversed
- By a set of trajectories (to one or more goals) previously found and traversed
- By a set of trajectories (to one or more goals) previously found and not traversed
- By the totality of (set of all possible) trajectories
- By a set of trajectories executed in the space in a random way

One can see that this knowledge contains implicitly both the description of the environment and the description of the actions required to traverse a trajectory in this environment. Moreover, if some particular system is the source of knowledge, then the collected knowledge contains information about properties of the system which moved in the environment.

All this information arrives in the form of experiences which record states, actions between each couple of states, and evaluation of the outcome. The collection of information obtained in one or several of these ways forms knowledge of space (KS).

If the information base contains all tessellata of the space with all costs among the adjacent tessellata, it is usually called the representation. Thus the representation can be considered equivalent to the multiplicity of explanations of how to traverse, or how to move. All kinds of learning, mentioned above, are equivalent: they belong to the same potential database reflecting reality exhaustively.

Links Between Planning and Learning

Planning is learning from experience in the domain of imagination: searching in the state space is exploration of these imaginary experiences. Planning is performed by searching within a limited subspace

- For a state with a particular value (designing the goal)
- For a string (a group) of states connecting SP and GP satisfying some conditions on the cumulative cost (planning of the course of actions)

The process of searching is associated either with collecting the additional information about experiences, or with extracting from KS the implicit information about the state and moving from state to state, or learning. In other words, planning is inseparable from and complementary to learning. Learning is a source of the multiscale (multiresolutional, multigranular) representation. Figure 4 illustrates how the multiscale representation emerges by consecutive generalization of the experiences. On the contrary, planning presumes consecutive refinement of the imaginary experiences. For both generalization and refinement, a set of procedures is used including grouping (G), focusing attention (FA), and combinatorial search (CS), which are together denoted GFACS.

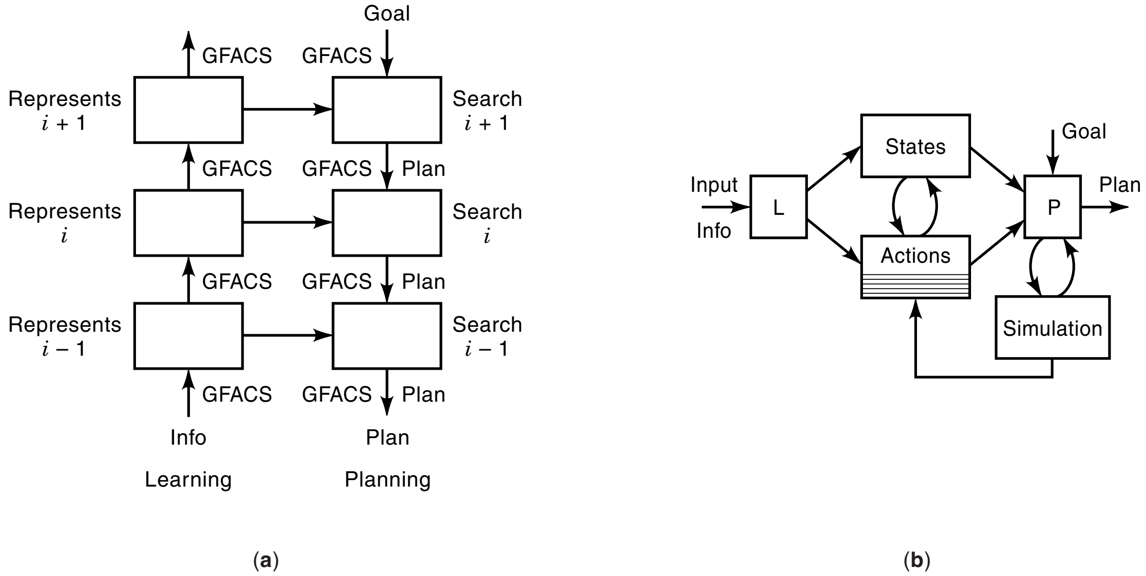

Figure 4. On the relations between planning and learning: functioning of GFACS in the joint learning–planning process. (a) Learning in a hierarchy, (b) learning at a level.

This unified planning/learning process is always oriented toward improvement of functioning in engineering systems (improvement of accuracy in an adaptive controller) and/or toward increasing the probability of survival (emergence of the advanced viruses for the known diseases that can resist various medications, e.g., antibiotics). Thus this joint process can be related to a system as well as to populations of systems, and determines their evolution.

PLANNING IN ARCHITECTURES OF BEHAVIOR GENERATION

Hierarchical Multiresolutional Organization of Planning

An important premise for introducing multiscale algorithms of planning is organization of a multiscale (multiresolutional, multigranular) world model. It is presumed that each system can be represented as a multiscale model, that is, as a hierarchy of models that differ in their degree of detail. This will allow for planning and control at each level of resolution (19). The multiscale world model, as well as multiscale system of planning/control modules, requires consecutive bottom-up generalization of the available information. Levels of generalization and the overall multiscale representation, as discussed here, are considered to be depictions of the same object with different degrees of accuracy. The preceding statement is given in mathematical form by applying concepts of the single-level state-space representation for the (not necessarily linear time-invariant) system (63–65):

$$x(t) = \mathrm{A}(x, u, t) \times (t) + B(x, u, t)u(t)$$
$$y(t) = \mathrm{C}(x, u, t) \times (t)$$

where

$$x \in R^n, \qquad u \in R^m, \qquad y \in R^p, \qquad t \in R^+$$

Thus it is possible to form a solution of these equations as mappings describing the state transition and output functions:

$$\Phi : R^n \times R^m \times R^+ \to R^n \times R^+$$
$$\Psi : R^n \times R^m \times R^+ \to R^p \times R^+$$

so that for any input function "u" on the interval $[t_0, t_f]$ it is possible to determine the corresponding output function "y" on the same interval. If it can be shown that there exists a pair of functions

$$\Phi' : R^{n'} \times R^{m'} \times R^+ \to R^{n'} \times R^+$$
$$\Psi' : R^{n'} \times R^{m'} \times R^+ \to R^{p'} \times R^+$$

for which n' is strictly less than n, and for which the same input function "u" generates the output function "y" such that inequality

$$\left| \int_{t_0}^{t_f} [y'(t) - y(t)]\,dt \right| < \epsilon$$

holds for all admissible inputs in the input function space where ϵ is a value which depends on the level of resolution under consideration. Then it is claimed that

$$\{\Phi', \Psi'\} \qquad \text{is an } \epsilon\text{-generalization of} \qquad \{\Phi, \Psi\}$$

The strictness of this formulation may be relaxed by considering a stochastic measure for associating a confidence level with the generalization to construct the concept of ϵ-generalization *nearly everywhere.* Thus

$$P\left[\left\| \int_{t_0}^{t_f} [y'(t) - y(t)]\,dt \right\| < \epsilon \right] < \tau$$

is a statement of the belief that the constraint holds with a probability defined by the preassigned threshold τ.

This formulation can be extended to an ordered collection of k epsilons $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_k\}$, thereby defining a hierarchy of k models which describe the same input-output behavior with increasing degrees of accuracy. The necessity of considering all elements of the input and output vectors as time varying functions may also be relaxed so that at some level 'i', u_{k_i} $[t_0, t_f]$ could be considered constant in the interval, whereas at some lower level (at higher resolution) the same input may be represented as a time-varying function.

The ability to formulate the world models with this hierarchical generalization will be shown in the following example to be an essential device for coping with the complexity associated with the planning of system operation in a combined feedforward–feedback controller.

Results of Planning the Path of the Vehicle

Figure 5 depicts the process of planning via snapshots of the screens presented to the user during planning. The process of search is shown in the left part of Fig. 5. The upper part shows the search in full space at low resolution. The lower part shows search in a reduced search space but at higher resolution. The final trajectory of the vehicle is shown in the

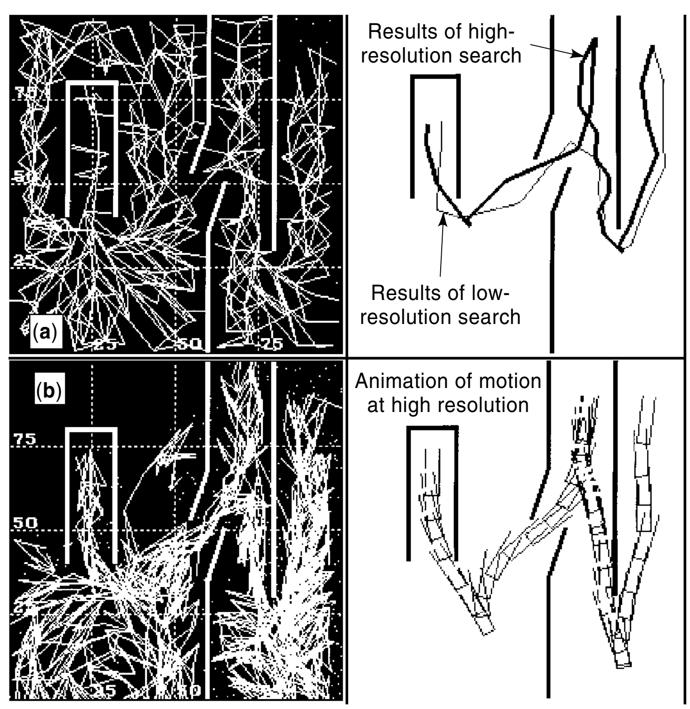

Figure 5. Planning via search in a multiresolutional state space. (a) Search in the whole space at low resolution, (b) search in the reduced space at high resolution.

right part of the figure for a workspace including a garage, wall, and gate.

The order of synthesis of this result can be seen beginning with Fig. 5, which is a depiction of the search tree at the low level of resolution, overlaid on the description of the workspace. The kinematics of the vehicle are clearly absent from this consideration as can be seen by the result of search at the first level in the upper right part of Fig. 5 (the thin-line trajectory), but are evident in the bold-line trajectory in the same figure, which is the result of search at the next level: one can see the maneuvering of the vehicle. The search at this high-resolution level is depicted in the lower parts of Figs. 4 and 5, where the reduced search tree of that level is shown.

This sequence of figures demonstrates that it is possible to synthesize complex maneuvers such as reversing and K-turns without using expert rule-base generated by a human being. Comparatively complex maneuvering is performed just by constructing a hierarchical representation of the system and searching for successive approximations to construct an ϵ-optimal solution of the problem.

Other Components of Planning

Planning consists of job assignment and scheduling. Job assignment distributes the motion among the spatial coordinates. Scheduling distributes the motion along the time axis. Together, they contribute to the search process. Search is performed by constructing feasible combinations of the states within a subspace (feasible, means: satisfying a particular set of conditions). Search is interpreted as exploring (physically, or in simulation) as many as possible alternatives of possible motion and comparing them afterward.

Each alternative is created by using a particular law of producing the group of interest (cluster, string, etc.). Usually, grouping presumes exploratory construction of possible combinations of the elements of space (combinatorial search) and as one or many of these combinations satisfy conditions of "being an entity"—substitution of this group by a new symbol with subsequent treating it as an object (grouping.)

The larger the space of search is the higher is the complexity of search. This is why a special effort is allocated with reducing the space of search. This effort is called focusing attention and it results in determining two conditions of searching, namely, its upper and lower boundaries:

1. Upper boundaries of the space in which the search should be performed
2. Resolution of representation (the lower boundaries)

Planning and Intelligence

Formation of multiple combinations of elements (during the search procedure, S) satisfying required conditions of transforming them into entities (grouping, G) within a bounded subspace (focusing attention, F) is a fundamental procedure in both learning and planning. Since these three procedures work together they can be considered as a triplet of computational procedures which include grouping, focusing attention and search (see GFACS in Fig. 4). Notice that in learning it creates lower resolution levels out of higher resolution levels (bottom-up) while in planning it progresses from the lower resolution levels out of higher resolution levels (top-down.)

This triplet of computational procedures is characteristic for intelligence and probably is the elementary computational unit of intelligence. Its purpose is transformation of large volumes of information into a manageable form which ensures success of functioning. The way it functions in a joint learning-planning process explains the pervasive character of hierarchical architectures in all domains of activities.

The need in GFACS is stimulated by the property of knowledge representations to contain a multiplicity of alternatives of space traversal (which is a property of representations to be redundant). Redundancy of representations determines the need in GFACS: otherwise the known systems would not be able to function efficiently (it is possible that redundancy of representations is a precondition for the possibility of life and the need in intelligence).

BIBLIOGRAPHY

1. A. Newell and H. A. Simon, GPS: A program that simulates human thought, in E. A. Feigenbaum and J. Feldman (eds.), *Computers and Thought,* R. Oldenbourg K. G., Cambridge, MA: MIT Press, 1963, pp. 279–293.
2. K.-S. Fu, Learning control systems, in J. T. Tou (ed.), *Advances in Information System Sciences,* New York: Plenum Press, 1969.
3. G. Saridis, *Self-Organizing Control of Stochastic Systems,* New York: Dekker, 1977.
4. G. Saridis and A. Meystel (eds.), *Proc. IEEE Workshop Intell. Control,* Troy, NY, 1985.
5. R. E. Fikes and N. Nilsson, STRIPS: A new approach to the application of the theorem proving to problem solving, *Artif. Intell.,* **2**: 189–208, 1971.
6. R. E. Fikes et al., Learning and executing generalized robot plans, *Artif. Intell.,* **3**: 251–288, 1972.
7. P. Hart et al., A formal basis for the heuristic determination of minimum cost paths, *IEEE Trans. Sys. Sci. Cybern.,* **SSC-4**: 100–107, 1968.
8. J. E. Doran and D. Michie, Experiments with the graph-traverser program, *Proc. Roy. Soc.,* **A**: 235–259, 1966.
9. W. E. Howden, The sofa problem, *Comput. J.,* **11**: 299–301, 1968.
10. T. Lozano-Perez and M. A. Wesley, An algorithm for planning collision free paths among polyhedral obstacles, *Commun. ACM,* **22**: 560–570, 1979.
11. J. S. Albus, Mechanisms of planning and problem solving in the brain, *Math. Biosci.,* **45**: 247–293, 1979.
12. T. Lozano-Pérez, Automatic planning of manipulator transfer movements, *IEEE Trans. Syst. Man Cybern.,* **11**, 1981.
13. M. Julliere et al., A guidance system for a mobile robot, *Proc. 13th Int. Symp. Industrial Robots,* Vol. 2, April 17–21, 1983, pp. 17–21.
14. R. Chavez and A. Meystel, Structure of intelligence for an autonomous vehicle, *Proc. IEEE Int. Conf. Robot. Autom.,* 1984, pp. 584–591.
15. J. E. Hopcroft et al., *SIAM J. Comput.,* **14** (2): 315–333, 1985.
16. A. M. Meystel, Planning in a hierarchical nested controller for autonomous robots, *Proc. IEEE 25th Int. Conf. Decision Control,* 1986, pp. 1237–1249.
17. J. C. Latombe, *Robot Motion Planning,* Boston: Kluwer, 1991.
18. M. Brady et al., *Robot Motion: Planning and Control,* Cambridge, MA: MIT Press, 1982.
19. J. S. Albus and A. Meystel, *Behavior Generation in Intelligent Systems,* Gaithersburg, MD: NIST, 1997.

20. R. W. Brockett, Poles, zeros, and feedback: State space interpretation, *IEEE Trans. Auto. Cont.,* **AC-10**: 129–135, 1965.
21. F. Albrecht et al., Path controllability of linear input-output systems, *IEEE Trans. Auto. Cont.,* **31**: 569–571, 1986.
22. H.-W. Wohltmann, Function space dependent criteria for target path controllability of dynamical systems, *Int. J. Control,* **41**: 709–715, 1985.
23. U. Kotta, Right inverse of a discrete time non-linear system, *Int. J. Control,* **51**: 1–9, 1990.
24. H. Seraji, Design of feedforward controllers for multivariable plants, *Int. J. Control,* **46**: 1633–1651, 1987.
25. K. Goldberg et al. (eds.), *Algorithmic Foundations of Robotics,* Wellesley, MA: A.K. Peters, 1995.
26. J. C. Latombe, Controllability, recognizability, and complexity issues in robot motion planning, *Proc. 36th Annu. Symp. Foundations Comput. Sci.,* 1995, pp. 484–500.
27. V. Akman, *Unobstructed Shortest Paths in Polyhedral Environments,* Berlin: Springer-Verlag, 1987.
28. R. E. Tarjan, A unified approach to path problems, *J. Assoc. Comp. Mach.,* **28**: 577–593, 1981.
29. A. Meystel and E. Koch, Computation simulation of autonomous vehicle navigation, *Proc. IEEE Int. Conf. Robot. Autom.,* 1984, pp. 159–169.
30. D. Keirsey et al., Autonomous vehicle control using AI techniques, *IEEE Trans. Softw. Eng.,* **11**: 986–992, 1985.
31. M. Montgomery et al., Navigation Algorithm for a Nested Hierarchical System of Robot Path Planning Among Polyhedral Obstacles, *Proc. IEEE Int. Conf. Robot. Autom.,* Raleigh, NC, 1987, pp. 1616–1622.
32. N. Rowe and R. Richbourg, An efficient Snell's Law method for optimal path planning across multiple two-dimensional irregular homogeneous-cost regions, *Int. J. Robot. Res.,* **9**: 48–66, 1990.
33. J. C. Latombe, A Fast Path Planner for a Car-Like Indoor Mobile Robot, *Proc. 9th Nat. Conf. Artif. Intell.,* Anaheim, CA, July 1991, pp. 659–665.
34. J. Barraquand and J. C. Latombe, Nonholonomic multobody mobile robots: Controllability and motion planning in the presence of obstacles, *Algorithmica,* **10** (2-3-4): 121–155, 1993.
35. P. Burlina, D. DeMenthon, and L. S. Davis, Navigation with uncertainty: Reaching a goal in a high collision risk region, *Proc. IEEE Int. Conf. Robot. Autom.,* 1992, pp. 2440–2445.
36. J. Borenstein and Y. Koren, Histogramic in-motion mapping for mobile robot obstacle avoidance, *IEEE Trans. Robot. Autom.,* **7**: 535–539, 1991.
37. R. Cerulli et al., The auction technique for the sensor based navigation planning of an autonomous mobile robot. *J. Intell. Robot. Syst.,* **21**: 373–395, 1998.
38. H. Takeda et al., Planning the motions of a mobile robot in a sensory uncertainty field, *IEEE Trans. Pattern Anal. Mach. Intell.,* **16**: 1002–1017, 1994.
39. A. Lazanas and J. C. Latombe, Motion planning with uncertainty: A landmark approach, *Artif. Intell.,* **76**: 285–317, 1995.
40. R. G. Simmons, Structured control for autonomous robots, *IEEE Trans. Robot. Autom.,* **10**: 34–43, 1994.
41. J. C. Latombe and C. Tomasi, An intelligent observer, C. Becker, H. Gonzalez-Banos, in O. Khatib and J. K. Salisbury (eds.), *Lecture Notes in Control and Information Sciences,* Vol. 223, New York: Springer, 1997, pp. 153–160.
42. S. M. LaValle et al., Motion strategies for maintaining visibility of a moving target, *Proc. 1997 IEEE Int. Conf. Robot. Autom.,* 1995, pp. 731–736.
43. K. S. Al-Sultan and M. D. S. Aliyu, A new potential field-based algorithm for path planning, *J. Intell. Robot. Syst.,* **17**: 265–282, 1996.
44. J.-O. Kim and P. K. Khosla, Real-time obstacle avoidance using harmonic potential functions, *IEEE Trans. Robot. Autom.,* **8**: 338–349, 1992.
45. J. Barraquand et al., Numerical potential field techniques for robot path planning, *IEEE Trans. Syst., Man, Cybern.,* **22**: 224–241, 1992.
46. C. Hein and A. Meystel, A genetic technique for planning a control sequence to navigate the state space, *1994 Goddard Conf. Space Appl. AI,* Greenbelt, MD: NASA, 1994, pp. 113–120.
47. K. Sugihara and J. Smith, *A Genetic Algorithm for 3-D Path Planning of a Mobile Robot,* Tech. Rep., Dept. Inf. Comput. Sci., Univ. Hawaii at Manoa, September 1996.
48. G. Grevera and A. Meystel, Searching in a multidimensional space, *Proc. 5th IEEE Int. Symp. Intell. Control,* Vol. 2, Philadelphia, PA, 1990, pp. 700–705.
49. C.-F. Liaw and C. C. White III, A heuristic search approach for solving a minimum path problem requiring arc cost determination, *IEEE Trans. Syst. Man Cybern.,* Part A, **26**: 545–551, 1996.
50. R. Bhatt et al., A real-time guidance system for an autonomous vehicle, *Proc. IEEE Int. Conf. Robot. Autom.,* Raleigh, NC, 1987, pp. 1785–1791.
51. R. Bhatt et al., A real-time pilot for an autonomous robot, *Proc. IEEE Int. Symp. Intell. Control,* Philadelphia, PA, 1987, pp. 135–139.
52. A. M. Meystel, Theoretical foundations of planning and navigation for autonomous robots, *Int. J. Intell. Syst.,* **2**: 73–128, 1987.
53. J. Albus, Outline for a theory of intelligence, *IEEE Trans. Syst., Man. Cybern.,* **21**: 473–509, 1991.
54. A. Meystel, *Autonomous Mobile Robots: Vehicles with Cognitive Control,* Singapore: World Scientific, 1991.
55. J. Barraquand et al., A random sampling scheme for path planning, *Int. J. Robot. Res.,* **16**: 759–774, 1997.
56. T. Skewis and V. Lumelsky, Experiments with a mobile robot operating in a cluttered unknown environment, *Proc. IEEE Int. Conf. Robot. Autom.,* 1992, pp. 1482–1487.
57. K. Sun and V. Lumelsky, Path planning among unknown obstacles: The case of a three-dimensional Cartesian arm, *IEEE Trans. Robot. Autom.,* **8**: 776–786.
58. V. Lumelsky, S. Mukhopadhyay, and K. Sun, Dynamic path planning in sensor-based terrain acquisition, *IEEE Trans. Robot. Autom.,* **6**: 462–472, 1990.
59. M. Kircanski and M. Vukobratovic, Contribution to control of redundant robotic manipulators in an environment with obstacles, *Int. J. Robot. Res.,* **5** (4): 112–119, 1986.
60. A. A. Maciejewsky and C. Klein, Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, *Int. J. Robot. Res.,* **4** (3): 109–116, 1986.
61. R. G. Roberts and A. A. Maciejewski, Nearest optimal repeatable control strategies for kinematically redundant manipulators, *IEEE Trans. Robot. Autom.,* **8**: 327–337, 1992.
62. T. S. Wilkman and W. S. Newman, A fast, on-line collision avoidance method for a kinematically redundant manipulator based on reflex control, *Proc. IEEE Int. Conf. Robot. Autom.,* 1992, pp. 261–266.
63. A. M. Meystel, Multiscale systems and controllers, *Proc. IEEE/IFAC Joint Symp. Computer-Aided Control Syst. Design,* Tucson, AZ, 1992, pp. 13–26.
64. A. Meystel and S. Uzzaman, Planning via search in the input–output space, *Proc. IEEE Int. Symp. Intell. Control,* Chicago, 1993, pp. 548–553.
65. J. Albus, A. Meystel, and S. Uzzaman, Nested motion planning for an autonomous robot, *Proc. IEEE Regional Conf. Aerospace Control Sys.,* Westlake Village, CA, 1993.

ALEX MEYSTEL
Drexel University

ROBOT PLANNING. See PLANNING.