Exact Robot Navigation Using Artificial Potential Functions

Elon Rimon, *Member, IEEE,* and Daniel E. Koditschek, *Member, IEEE*

***Abstract*—We present a new methodology for exact robot motion planning and control that unifies the purely kinematic path planning problem with the lower level feedback controller design. Complete information about the freespace and goal is encoded in the form of a special artificial potential function—a *navigation function*—that connects the kinematic planning problem with the dynamic execution problem in a provably correct fashion. The navigation function automatically gives rise to a bounded-torque feedback controller for the robot's actuators that guarantees collision-free motion and convergence to the destination from almost all intial free configurations. Since navigation functions exist for any robot and obstacle course, our methodology is completely general in principle. However, this paper is mainly concerned with certain constructive techniques for a particular class of motion planning problems. Specifically, we present a formula for navigation functions that guide a point-mass robot in a *generalized sphere world.* The simplest member of this family is a space obtained by puncturing a disc by an arbitrary number of smaller disjoint discs representing obstacles. The other spaces are obtained from this model by a suitable coordinate transformation that we show how to build. Our constructions exploit these coordinate transformations to adapt a navigation function on the model space to its more geometrically complicated (but topologically equivalent) instances. The formula that we present admits sphere-worlds of arbitrary dimension and is directly applicable to configuration spaces whose forbidden regions can be modeled by such generalized discs. We have implemented these navigation functions on planar scenarios, and simulation results are provided throughout the paper.**

I. Introduction

IN this paper we present a technique for constructing artificial potential-fields[1] that are guaranteed to bring a bounded-torque actuated robot to a desired configuration without its hitting obstacles in a perfectly known and stationary environment. In order to guarantee the correctness of the feedback controllers resulting from these constructions, we have found it necessary to refine the notion of "artificial potential field" by adding several new technical requirements. We emphasize the importance of these refined conditions by referring to potential functions that meet them as *navigation functions.* In particular, our navigation functions have only one minimum—the goal configuration.

The notion of a navigation function at once combines a solution to the global findpath problem along with a feedback controller for the robot. This approach, therefore, promises to free the "higher level" planner to a more abstract level. Indeed, since we have shown that navigation functions always exist, these ideas offer, in principle, a completely general solution to the geometric piano movers problem. However, the constructive techniques presented in this paper encompass a greatly restricted class of problems.

The construction we will present applies to a point-mass robot moving in essentially any *generalized sphere world.* This is any space resulting from the application of a suitable coordinate transformation to the *model sphere world*—a disc punctured by an arbitrary number of smaller disjoint discs representing obstacles. Thus, our present recipes work for any n-degree-of-freedom (DOF) robot whose configuration space happens to be a generalized sphere world. For instance, almost any 2-DOF robot and obstacle course give rise to a configuration space each of whose path-connected components is a two-dimensional generalized sphere world. Of course, the techniques presented here may well turn out to offer reasonable heuristic strategies in many other situations. In fact, we have recently used them to construct navigation functions for a certain class of rigid bodies [36] and for an assembly problem [23]. Such considerations fall, however, outside the scope of the present paper, which is concerned exclusively with the exact, provably correct aspect of the methodology as applied to generalized sphere worlds.

This section continues with a description of the navigation problem and the potential-field methodology. The next section briefly reviews a number of important preliminary ideas and results that have been previously presented in more mathematically oriented literature [25], [38]. Section III contains the central contribution of the paper: a systematic account of how to construct navigation functions from a geometric description of the free configuration space. Section IV assesses the computational complexity of the resulting navigation algorithms—they turn out to be comparable in this regard to the more traditional findpath solution techniques—and presents a sampling of the many simulation studies we have made in the course of this research. The paper concludes with a brief summary and a suggestion of what we consider to be the more promising areas for extension of these ideas.

Manuscript received March 19, 1991; revised December 11, 1991. This work was supported in part by the National Science Foundation under Grant DMC-8505160 and by a Presidential Young Investigator Award. Portions of this paper were presented at the IEEE International Conference on Robotics and Automation, Cincinnati, OH 1990.

E. Rimon was with the Center for Systems Science, Department of Electrical Engineering, Yale University, New Haven, CT 06520. He is now with the Computer Science Department, Stanford University, Palo Alto, CA 94305.

D. E. Koditschek is with the Center for Systems Science, Department of Electrical Engineering, Yale University, New Haven, CT 06520.

IEEE Log Number 9200371.

[1] The term *potential field* means the negated gradient vector field $-\nabla V$ of a potential function V.

1042-296X/92/$03.00 © 1992 IEEE

A. Problem Statement

Consider an n-DOF robot that performs various navigational tasks in a world populated by obstacles. In its simplest form the navigation problem can be stated as follows: The robot presented with a fixed destination q_d must approach the destination and halt there while avoiding the obstacles. As stated above, this paper is concerned with solutions that yield a bounded-torque controller from perfect information about a stationary world. All of these terms require more careful explanation.

The possible placements of the collection of n robot links in the real world can be described as points in the n-fold cross product of the Euclidean group (all translations and rotations). The *configuration space* $\mathcal{CS}$ is the n-dimensional subset of this space corresponding to placements that respect the robot's joint constraints. The *joint space* $\mathcal{J} = E^n$ (E^n is a Euclidean n-space) is a special local coordinate system on $\mathcal{CS}$, having the property that each coordinate corresponds to a physical joint. The *phase space* $\mathcal{T}(\mathcal{CS})$ is similarly represented locally by the *joint phase space,* $\mathcal{TJ} = E^n \times E^n$ of joint positions and velocities. We presume that the robot's dynamical behavior can be satisfactorily described by the standard Lagrangian model (ignoring such nonideal phenomena as stiction, gear backlash, flexibility, etc.) Further, we assume the robot has ideal sensors that measure its position and velocity accurately and without delay, and that its actuators instantaneously deliver any specified torque τ whose magnitude is bounded. Thus, our *robot model* takes the following commonly accepted description in $\mathcal{TJ}$:

$$[M(p)]\ddot{p} + f(p, \dot{p}) + g(p) = \tau \tag{1}$$

where $M(p)$ is the robot's inertia matrix, $f(p, \dot{p})$ and $g(p)$ are fictitious and gravitational forces, and τ is the torque input. As noted below, one distinguishing virtue of the potential-field method is that the specifics of M and f need not be known.

The *free configuration space* $\mathcal{F} \subset \mathcal{CS}$ is the subset of $\mathcal{CS}$ obtained by removing all configurations involving intersection of the robot with physical obstacles or intersection between the robot's links. The regions removed from $\mathcal{CS}$ are referred to as *configuration-space obstacles.* Note that another kind of forbidden region whose definition involves the allowed initial velocities at each initial free configuration is imposed by the boundedness of the input torques. This constraint is specified in phase space $\mathcal{T}(\mathcal{CS})$ and will be discussed in Section II.

By perfect information we mean a complete description of the configuration-space obstacles, certain other geometric constants that may be derived from them, as well as a topological model of the resulting freespace $\mathcal{F}$. We do not detail the computational steps required to produce the derived constants in any generality—a complete list that we shall refer to as the *geometric data* throughout the sequel is assumed to be *a priori* known and is merely appended to the paper.[2] As for a topological model, all the configuration spaces in this paper will be the *generalized sphere worlds* mentioned above.

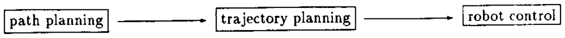


Fig. 1. The traditional three-step decomposition of the navigation problem.

The prospects for developing constructive solutions to more general versions of this problem that do not impose excessively strong requirements for *a priori* information will be touched upon in Section I-B-2.

We may now state our version of the navigation problem: Given the geometric data and a robot model, find a neighborhood of initial velocities about $\mathcal{F}$ in phase space $\mathcal{E}_c \subset \mathcal{T}(\mathcal{CS})$. Construct a smooth bounded-torque feedback controller $\tau(p, \dot{p})$ such that trajectories of the closed-loop robot system originating in $\mathcal{E}_c$ remain there for all future time and eventually approach the destination q_d and halt there.

Note that this formulation avoids the question of whether q_d is reachable from some particular initial configuration or not. Strictly speaking, our method does not provide a means of making this determination, as will be detailed in Section I-B-2. In practice, assuming the parameter values have been adequately adjusted, we may conclude with probability one that the destination is not reachable if the robot's motion ceases short of the goal.

B. Comparison with Alternative Methodologies

Since many researchers have addressed diverse variations on the navigation problem over the last decade, it seems important to sketch the place of our potential-field method in the more general literature. This will provide at the same time an opportunity to speculate briefly on the prospects for generalizing our present results.

The robot navigation problem has traditionally been decomposed into the three separate steps depicted in Fig. 1. First, the *path planning* problem is solved: given geometric data and desired destination, a collision-free curve connecting the robot's initial and final configurations is constructed in the free configuration space $\mathcal{F}$. This curve completely ignores the robot's dynamics. Second, the *trajectory planning* problem is solved: given a collision-free curve in $\mathcal{F}$, find a time parameterization for the curve under certain constraints. The resulting time-parameterized curve becomes a reference trajectory, $q_r : [0, T] \to \mathcal{F}$. Third, the attendant *control problem* is solved: a control scheme is devised to make the physical robot follow the reference trajectory as close as possible.

In contrast, the potential-field approach unifies these three steps into one. Consider a real-valued map constructed on the robot's free configuration space, $V : \mathcal{F} \to R$, which has a unique minimum at q_d, the goal configuration, and is uniformly maximal over the boundary of $\mathcal{F}$ (q_d must of course be specified in the interior of $\mathcal{F}$). For example, V on a planar configuration space can be visualized as a "sculptured" two-dimensional surface constructed over $\mathcal{F}$ (see Fig. 2). Subject to few additional technical requirements listed in Section II-A, such maps constitute the class of *navigation functions.*[3] Once a suitable V is constructed it automatically determines

[2] Since these derived constants are used solely to compute lower bounds on certain key parameters in our constructions, we have found little sensitivity in practice to inaccurate estimates. For example, in all of our simulation studies, the parameters have been simply set "very high."

[3] Note, however, that the scope of our constructions is much broader than the sample picture suggests. We show how to build such functions on a class of Euclidean spaces of arbitrary dimension.

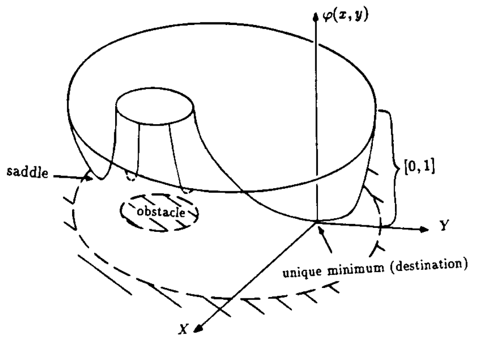


Fig. 2. A suitable potential function over a free configuration space.

a feedback control law of the form

$$\tau(p,\dot{p}) = -\nabla V(p) + d(p,\dot{p}) \qquad (2)$$

where d is an arbitrary dissipative vector-field.[4] It is shown in [24] that the critical qualitative behavior of V's gradient trajectories (i.e., solutions of $\dot{p} = -\nabla V(p)$) is copied by the robot itself. Since V has a unique minimum at q_d and is maximal over the boundary of $\mathcal{F}$, almost all the gradient trajectories of V approach q_d without touching the boundary of $\mathcal{F}$ (global convergence is precluded by the topology of $\mathcal{F}$ [25]). Consequently, subjecting the physical robot to the feedback (2) causes the closed-loop robot system to "inherit" the same behavior. Namely, the robot approaches q_d while avoiding the obstacles. Thus, a suitable V constitutes a path-planning algorithm, a trajectory-planning algorithm, and specifies a feedback control law, all at once.

We now briefly review some of the representative contributions from the alternative literature addressing the navigation problem before proceeding to compare these methods with the potential-field approach.

1) Alternative Literature: The purely geometric problem of constructing a collision-free path in a space cluttered with polynomially described obstacles (given perfect information) has been solved by Schwartz and Sharir [39]. Moreover, a near-optimal solution has been offered by Canny [9]. Schwartz and Sharir's algorithm uses a "cell decomposition" approach. It consists of first decomposing the free configuration space into a finite collection of cells and then searching the adjacency graph of these cells. Canny's algorithm constructs a similar graph called a "roadmap" for $\mathcal{F}$, with a much more efficient computational tool (multivariate resultant). The cost of constructing the roadmap is polynomial in the number of semialgebraic constraints used to describe $\mathcal{F}$ and in the degree of these constraints. It also tends to grow exponentially with the robot's degrees of freedom.

The (traditional) path-planning algorithms described above specify a collision-free curve in the free configuration space $\mathcal{F}$. They do not take into account the robot's timing or velocity. One requires a parameterization of the given path by time since the (traditional) controller discussed in the next paragraph expects as input a reference trajectory of the form $q_r : [0,T] \to \mathcal{F}$, where the image of q_r is the collision-free curve. For $q_r(t)$ to be feasible, the parameterization must respect the bounds on the torque generated by the robot's actuators. Two such techniques, the "switching" (or bang-bang) technique [7] and "kinodynamic planning" [11], compute a time parameterization that minimizes the total travel-time T.

Traditionally, the attendant control problem assumes a given collision-free trajectory $q_r : [0,T] \to \mathcal{F}$ connecting the robot's initial and final configurations. The problem is to make the physical robot follow this path without its hitting obstacles and without exceeding the actuators' torque limits. The most commonly used exact method is called the "computed torque," or "inverse dynamics" technique [13], [29]. It consists of exact term-by-term cancellation of the noninlearities in the robot's dynamics (see (1)), by using

$$\tau(p,\dot{p}) = f(p,\dot{p}) + [M(p)\tau_1] \qquad (3)$$

as control input where τ_1 is not yet specified. The resulting closed-loop system

$$\ddot{p} = \tau_1 \qquad (4)$$

is linear, and the standard control input used to achieve tracking of $q_r(t)$ is

$$\tau_1(p,\dot{p}) = \ddot{q}_r - K_2(\dot{p} - \dot{q}_r) - K_1(p - q_r) \qquad (5)$$

where K_1 and K_2 are positive definite (symmetric) matrices. The resulting closed-loop linear system obtained by applying (5) to (4) converges asymptotically to $q_r(t)$.

2) Generality and Practicability of the Potential Fields: We shall see in Section II that navigation functions exist in principle on each connected component of practically any free configuration space. This paper provides, however, a constructive recipe only for a restricted class of robot configuration spaces—the n-dimensional generalized sphere worlds. The computational cost of this recipe turns out to be polynomial in the number of degrees of freedom (see Section IV)[5] and polynomial in the number of "geometric features." In contrast, the findpath algorithms mentioned above are both general and constructive (although we are not familiar with any practical implementations). On the other hand, their computational cost is exponential in the number of degrees of freedom. Of course, the relative efficiency of the potential-field approach as applied to the sphere worlds can only result from our starting with relatively greater information.[6]

The classical findpath algorithms presume *a priori* knowledge of the robot's geometry and that of the environment. In contrast, an extrapolation of our present solution techniques to the general problem would impose an excessively strong

[4] In order to simplify the exposition, assume the absence of gravitational forces.

[5] As discussed in Section IV-A, this count does not include the computation of certain geometric constants that are assumed to be part of the geometric data throughout the paper.

[6] We are presently unsure whether this relative efficiency is a general consequence of the greater amount of *a priori* information we require—a topological model together with certain additional geometric data—or simply an artifact resulting from our representation of generalized sphere worlds as piecewise quadratic shapes.

version of perfect information—knowledge of a *topological model* for the free configuration space. This entails, for instance, *a priori* knowledge of whether a path exists to the destination or not. The topological characterization of the free configuration space does not seem to have been addressed in the literature, and there seems to be no indication what its computational cost would be. The need for a topological model is consequently the chief practical obstruction to extending our present constructive techniques beyond the cases presented here.

Yet it seems premature to judge the intrinsic limitations of the potential-field approach. For example, research presently in progress seems to indicate the possibility of adaptively adjusting the potential field in the course of navigation in order to do away with the formal requirement for any geometric information other than the location of the obstacles' boundaries. Moreover, the dependence of these constructions upon *a priori* topological models may to some degree be an artifact of our correctness proofs. For example, recent work by the first author demonstrates that our sphere-world navigation functions (reviewed in Section II-B-2) constitute navigation functions for the entirely unrelated topology arising from a class of planar rigid-body motion planning problems [36].

Whatever the liabilities, the strong version of the navigation problem as we have posed it presents some advantages. Computationally oriented solutions, the roadmap algorithm for instance, must be reinitialized each time the information changes. In contrast, our potential fields are given as a closed-form symbolic expression in terms of the obstacles' geometry and the robot's destination. This use of symbolic information allows the navigation function to change continuously (and relatively cheaply from the computational viewpoint) with changes in the available information.[7] In this sense, our longer term goal might be seen as a parameterized version of the "error detection and recovery" problem that has already been posed (and approached nonparametrically) by Lozano-Perez *et al.* [28] and Donald [12].

From the most practical point of view, controllers resulting from the potential-field approach have a long history of successful implementation in the robotics literature [17], [27], [31]. Moreover, there may be some theoretical reasons to prefer the "natural control" methodology of (2) to the inverse dynamics methods of (3) and (4). For example, some have argued that it presents a minimal energy solution to the obstacle-avoidance problem [41]. In contrast, the cancellation of all the nonlinear terms in the robot's dynamics according to the computed torque method reflects the designer's uncertainty as to which of these terms should be considered "undesirable." But it is never possible to cancel terms exactly, even using adaptive versions of the inverse dynamics scheme. Small cancellation errors may give rise to unacctably large tracking errors. Such tracking errors may also cause the robot to exceed its actuators' torque limits, affecting its ability to compensate for the tracking errors. In contrast, the natural control approach exploits the special properties of mechanical systems (of which the robot is a special class) to avoid such cancellations as much as possible.

C. Related Literature

We conclude this introductory section with a brief summary of other potential-field related research in robotics. Khatib, in his 1980 Ph.D. dissertation [16], proposed a closed-form expression for a potential function and suggested to use its negative gradient as torque input. Unfortunately, his potential function suffers from several problems, the most important one being undesired local minima. This phenomenon persists even with obstacles having simple shape (see [19] for example). Since Khatib's work a decade ago, numerous investigators have attempted to use potential functions in various robotic applications. These papers are described in detail in [35]. They roughly fall into three categories: 1) Papers addressing the problem of undesired local minima. For example, Krogh's generalized potentials [26], Connolly's Laplacian approach [10], and Barraquand's simulated annealing technique [5]. 2) Papers that view potential functions as path-planning aid. For example, the "valley tracking" technique [4] and Warren's path optimization procedure [45]. 3) Papers that extend the use of potential functions to time-varying situations. For example, Newman and Hogan's work in the context of avoiding moving obstacles [32] and Khatibs's real-time obstacle-avoidance technique [17].

None of the papers mentioned above offers an exact potential-function-based algorithm that is guaranteed to work.[8] The hardest problem to overcome is the presence of undesired local minima. This problem has been observed in many practical implementations (see [1], [8], [17], [43]). A second problem, encountered by those addressing the robot's dynamics, is the following seemingly contradictory requirement: in order to prevent collision with obstacles, it seems that the obstacles should exert an arbitrarily large repulsive potential as the robot approaches their boundary. At the same time, practical bounds on the robot's actuators impose a limit on the allowable repulsive potential. Yet docking, parts mating, and more general motion tasks [15], all require navigation at or along the boundary of the configuration space.

In contrast, the navigation functions, once constructed, provide a solution to both problems.

II. Preliminary Definitions and Results

Before proceeding to a presentation of the general constructions, it is helpful to briefly review a few ideas, and results from our earlier work. In Section II-A we motivate and define the notion of a navigation function as providing a solution to the problem statement of Section I-A above. In Section II-B, we present a catalog of "functional building blocks" that will be used systematically in the sequel.

[7] For example, there is nothing to impede the search for a navigation function that is affected only by obstacles in the vicinity of the robot. A preliminary investigation into this area is described in the concluding section.

[8] An exception is Connolly *et al.* [10] who propose the numerical integration of Laplace's equation, subject to constraints that characterize a navigation function.

A. Navigation Functions

This section reviews the fundamental limits in the potential-field approach arriving thereby at the notion of a navigation function. We first examine the problem of controlling the robot with potential fields. Two unremarkable observations by the second author [24]—the first reviewed in Section II-A-1, the second in Section II-A-2—afford a list of criteria that we desire of the potential fields. That examination leads to Definition 1, the cornerstone of our subsequent work in this area. The existence of functions satisfying this definition is guaranteed as discussed in Section II-A-3. We finally introduce the notion of invariance under coordinate transformation in Section II-A-4 that will be a key to the later constructions.

1) Admissible Potential Functions for Obstacle Avoidance: A century ago, Lord Kelvin [42] observed that a dissipative mechanical system—that is, (1) when τ is a the sum of a potential force and a dissipative force—loses total energy along its trajectories and that consequently trajectories originating near a potential-energy minimum would asymptotically approach it. In the robotics context, Khatib [18] proposed replacing the gravitational forces with an artificial potential field and then adding a dissipative term, as in (2).[9] Khatib's method provides a means of obstacle avoidance and local approach to the destination. This idea was subsequently and independently introduced in its full generality by a number of researchers [3], [21], [30], [33]—see [24] for a more detailed discussion of the history. To the best of our knowledge,[10] all of the previous proponents of this technique have required unbounded input torques at the obstacle boundaries. We shall see now that admissible potentials yield bounded controllers.

A real-valued function on the free configuration space is said to be *admissible* if is uniformly maximal on the boundary of $\mathcal{F}$, i.e., where the robot touches an obstacle

$$V(q) = \begin{cases} = c, & \text{for all } q \in \text{boundary } (\mathcal{F}) \\ < c, & \text{for all } q \in \text{interior } (\mathcal{F}). \end{cases}$$

for some constant $c > 0$ (see, e.g., Fig. 2). Using energy-conservation considerations, it can be shown that trajectories of a dissipative system with admissible potential energy that start in $\mathcal{F}$ with suitable initial velocity remain away from the obstacles [24]. Moreover, since any realistic robot configuration space is compact, any smooth potential field defined on it is automatically bounded. Thus, a potential-field controller obtained from an admissible function is bounded and steers the robot away from the obstacles if some initial speed limit is imposed on the robot.

In fact, this speed limit must be provided as a function of initial position: large but bounded speeds are tolerable from initial configurations away from the obstacles; initial configurations exactly on the obstacles' boundary must have zero velocity. For example, let U be an admissible function with a maximal height of unity. Let $K(q, \dot{q})$ be the robot's kinetic energy, and let $(q_0, \dot{q}_0)$ be its initial position and velocity. Choose a "gain" c according to the inequality

$$c \geq \frac{K(q_0, \dot{q}_0)}{1 - U(q_0)}.$$

Using the potential $V \triangleq cU$, it follows that the bounded-torque controller resulting from (2) steers the robot away from the obstacles toward some local minimum of V. This simple procedure can be extended to a guarantee that trajectories starting in the interior of $\mathcal{F}$ stay inside an explicitly specified subset of $\mathcal{F}$, thus providing for a safety clearance from the obstacles. A more careful discussion of how to adjust the height of V in order to meet magnitude constraints on the input torque is beyond the scope of this paper. Nevertheless, it is clearly forthcoming from these observations.

2) Morse Functions for Convergence: If few researchers have considered the possibility of avoiding infinite torques, seemingly none has heretofore inquired as to the possibility of obtaining global navigation results. One of the elementary results of dynamical systems theory establishes that if an unstable equilibrium is nondegenerate—that is, the linearized vector-field has full rank there—then its domain of attraction has an empty interior. For example, the dynamical system

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -x \\ y \end{pmatrix}$$

has an unstable equilibrium (a saddle) at the origin and its domain of attraction (the x axis in the (x, y) plane) has an empty interior.

A simple computation shows that the equilibrium states of the closed-loop system (1) coincide with the critical points of the potential function and that they are nondegenerate if and only if the Hessian (the matrix of second derivatives) of the potential function evaluated at its critical points is nonsingular. We will follow standard mathematical convention and refer to such functions as *Morse functions.* Another appeal to Lord Kelvin's total energy argument, now conceived of as a Liapunov function for the closed-loop system, establishes in consequence that, when the potential field is the gradient of a Morse function, almost all the initial states asymptotically approach one of the local minima. For details, refer to [24].

Although "essential" global convergence (that is, convergence from almost all initial configurations) is perfectly acceptable from the practical point of view, there remains the question of whether true global convergence is achievable. The answer is no. To begin with, elementary continuity arguments impose constraints: any continuous function V on a compact set $\mathcal{F}$ must attain its maximum and minimum on that set. It might be imagined, however, that a sufficiently clever construction could be "strongly" admissible in the sense that the only local maxima occur on the boundary and nowhere else. Indeed, our constructions do have this property. In any case, the appearance of interior saddle points is unavoidable. It can be shown under very general conditions that the set of initial states attracted by a dynamical system to an asymptotically stable equilibrium is a topologically deformed disk [6]. But the only free spaces with the topology of a disk are the trivial ones with no interior obstacles! Thus, in every other case there must

[9] Actually, Khatib proposed exact cancellation of all the nonlinearities due to the rigid body dynamics as well. The identical limit behavior obtains in either case. It is an open question as to which alternative results in more desirable transient behavior.

[10] That is, with the exception of Newman and Hogan [32] who proposed the use of bang-bang (discontinuous but bounded) controllers.

be additional equilibrium states.[11] The necessary appearance of spurious unstable equilibria need not cause any practical difficulties if it can be assured that "few" initial conditions will get stuck on them. If the potential function is a Morse function, then the preceding arguments provide exactly this assurance.

3) The Existence of Navigation Functions: Since almost all initial positions and velocities (with appropriate initial total energy) lead away from the obstacles and toward some minimum, an admissible Morse function with a single minimum at the desired destination provides a solution to the strong version of the robot navigation problem as posed in Section I-A. We emphasize the importance of these various conditions by giving them a formal name:

Definition 1 (from [25]): Let $\mathcal{F}$ be a robot free configuration space, and let q_d be a goal point in the interior of $\mathcal{F}$. A map $\varphi : \mathcal{F} \to [0,1]$ is a *navigation function* if it is

1) smooth on $\mathcal{F}$ (at least a $C^{(2)}$ function);
2) polar at q_d, i.e., has a unique minimum at q_d on the path-connected component of $\mathcal{F}$ containing q_d;
3) admissible on $\mathcal{F}$, i.e., uniformly maximal on the boundary of $\mathcal{F}$;
4) a Morse function.

The intuitive motivation for each of these items has been discussed above with the exception of the first. Smoothness ensures that $\nabla\varphi$ be continuously differentiable and consequently that the classical existence and uniqueness results of solutions for the closed-loop robot system apply. Moreover, the notion of a Morse function (Property 4) is well defined only for smooth functions. But, smoothness is related to another notion that Definition 1 does not address at all—the practical requirement that φ ought to be *effectively computable.* This notion seems harder to formalize. In previous work we have insisted not only on smoothness but on analyticity (the existence of a global Taylor series). We have in mind, of course, closed-form expressions in which a finite number of familiar analytic functions like polynomials are composed together by elementary algebraic operations. Often, however, real-world scenes give rise to free spaces that are merely semi-analytic, i.e., finite Boolean combinations of analytic shapes. One might expect that such situations will require nonanalytic functions as well (e.g., squareroots) and, indeed, this emerges in the constructions presented later in this paper. In general, we strive for "the most elementary" functions possible. Specifically, the constructions presented in this paper use simple algebraic combinations of polynomials and kth root functions.

When does a robot free configuration space admit a navigation function? Smale proved the generalized Poincaré's conjecture in higher dimensions roughly three decades ago. In so doing, he was led to develop a number of results concerning potential fields of which the most important for this investigation concerns the existence of navigation-like functions (Smale called such functions *nice* [40]) on spaces without boundary. Elementary results from algebraic topology may be used to extend Smale's ideas to the case of a boundary which arises naturally in the robotics context. We may thus assert the following general existence result [25].[12] For every smooth connected and compact manifold with boundary, $\mathcal{M}$, and any interior point, $x_0 \in \mathcal{M}$, there exists a smooth navigation function with a (unique) minimum at x_0.

One would, ideally, like to obtain a closed-form formula in terms of the geometric data and q_d for a navigation function on a completely general free configuration space $\mathcal{F}$. Unfortunately, the encouraging result that navigation functions always exist is not constructive, and there remains the mundane task of actually constructing them. This is the chief concern of this paper. Indeed, our research program concerns the actual construction of such navigation functions for increasingly more realistic robotic scenarios.

4) Navigation Properties Are Invariant Under Coordinate Transformation: A central tenet of the larger program of research within which our version of the navigation problem takes its place is the importance of generalization across a common task domain [22]. Simply put, we seek to use and reuse the same solution techniques by systematically identifying when two apparently different tasks are actually the same. Indeed, one of the principal reasons for formalizing the statement of a problem is the possibility of establishing such identities. In this section we explore such features of the navigation problem. Specifically, Definition 1 implies that two navigation problems no matter how they differ in geometric detail are actually the same if there exists a coordinate transformation mapping one to the other. Conversely, this implies that the navigation properties are invariant under transformation of both the domain and the range spaces.

Formally, a map between two spaces that is smooth, one-to-one and onto, and has a smooth inverse is called a *diffeomorphism.* In general, diffeomorphisms preserve the properties of navigation functions as the following two facts assert [25]. The first constitutes a guarantee for the existence of navigation functions over every space $\mathcal{F}$ that can be transformed to a given model space $\mathcal{M}$. Let $\mathcal{M}$ and $\mathcal{F}$ be two free configuration spaces. Let $\hat{\varphi} : \mathcal{M} \to [0,1]$ be a navigation function on $\mathcal{M}$, and let $h : \mathcal{F} \to \mathcal{M}$ be a diffeomorphism. Then

$$\hat{\varphi} = \sigma \circ h$$

is a navigation on $\mathcal{F}$. The second fact asserts that navigation properties are invariant under transformation of the range space as well. Let $\hat{\varphi} : \mathcal{F} \to \mathbb{R}$ and $\sigma : \mathbb{R} \to \mathbb{R}$ be smooth, and let φ be defined by

$$\varphi = \sigma \circ \hat{\varphi}.$$

If σ is monotonically increasing, then the set of critical points of $\hat{\varphi}$ and φ coincide, and the type of each critical point (i.e., minimum, maximum, or a saddle) is identical.

These facts will be used systematically below to enlarge the scope of the constructive results. For example, the first fact will

[11] We have shown in [25] even a more specific result: a smooth nondegenerate vector field on any generalized sphere world that has a unique attractor and that is inward directed on the boundary must have at least as many saddles as there are internal obstacles.

[12] It has been called to our attention that similar existence results might be obtained from more classical arguments involving partial differential equations with specified boundary constraints.

be used in Section III to extend our navigation functions from the "Euclidean sphere worlds"—a geometrically trivial class of obstacle courses—to almost any topologically deformed instance. The second will be used in Section III to modify a given cost function on $\mathcal{F}$ into a navigation function.

B. A Catalog of Preliminary Constructions

Having established the conceptual groundwork, it seems helpful to introduce some useful "functional building blocks" here. We will discuss in Section II-B-1 the manner in which the geometric data will be used in the constructions to follow. In Sections II-B-2 and II-B-3 we introduce a number of simple scalar functions that will be used to construct navigation functions. We shall use the notation $f_1 \circ f_2 = f_1(f_2(x))$ to denote function composition.

1) Implicit Representation of the Geometric Data: In the problem statement of Section I-A we assumed the knowledge of geometric data. Let us describe in detail how this information is used in the constructions to follow.

To begin with, we shall use the following distance-to-the-goal function, defined by

$$\gamma_\kappa(q) = \|q - q_d\|^{2\kappa} \tag{6}$$

where $\|\cdot\|$ is the Euclidean norm and $\kappa > 0$ is a parameter.

The various configuration-space obstacles will appear in the constructions to follow via their implicit representation. That is, each obstacle will be represented by an *obstacle function,* a real-valued map β_i representing it in the form[13]

$$\text{obstacle}_i = \{q : \beta_i(q) \leq 0\}.$$

It can be shown (pathological situations excluded) that β_i is zero exactly on the boundary of the obstacle in question.

For example, consider the Euclidean sphere worlds, as illustrated in Fig. 3. Let $\mathcal{D}(q, \rho)$ denote a Euclidean n-dimensional disk with center $q \in E^n$ and radius ρ. A *Euclidean sphere world* is formed by removing from a large n-dimensional disk, $D_0(0, \rho_0)$, M smaller disk-like "punctures," $D_j(q_j, \rho_j)$ for $j = 1 \cdots M$, called the *obstacles.* For simplicity, we shall regard the complement of $\mathcal{D}_0$ in E^n as the zeroth obstacle. The *free configuration space* remains after removing all the internal obstacles from $\mathcal{D}_0$

$$\mathcal{F} \triangleq \mathcal{D}_0 - \bigcup_{j=1}^{M} \text{obstacle}_j.$$

For $\mathcal{F}$ to be a valid sphere world, the obstacles' closure must be disjoint and be contained in the interior of $\mathcal{D}_0$. In this example, the *geometric data* takes the form of $M + 1$ centers q_i and radii ρ_i. From this data it is straightforward to construct the spherical obstacle functions:

$$\beta_0(q) = -\|q - q_j\|^2 + \rho_0^2$$
$$\beta_j(q) = \|q - q_j\|^2 - \rho_j^2, \qquad \text{for } j = 1 \cdots M.$$

More generally, we shall use formulas developed by Zenkin [46] for the implicit representation of a set described by an arbitrary Boolean combination of other sets. These formulas are expressed in terms of the implicit representation of the constituent shapes, which are assumed to be known. Specifically, if m sets in E^n are described by the inequalities $\beta_i(q) \leq 0$, then their intersection is given by $\psi_\cap(\beta_1, \cdots \beta_m) \leq 0$, and their union by $\psi_\cup(\beta_1, \cdots \beta_m) \leq 0$. These formulas are completely general, and we will use them to construct the implicit representation of complicated obstacles by taking Boolean union and intersection of simpler ones.

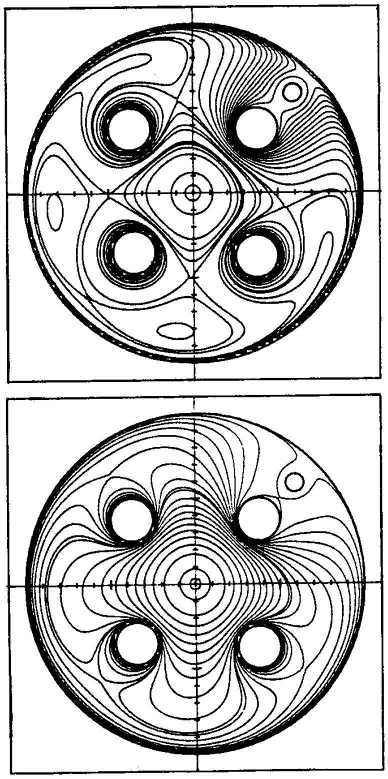


Fig. 3. Planar sphere world with five internal obstacles. The undesired local minima (top) disappear as the parameter value increases (bottom).

Finally, we will be typically interested in the product of all the obstacle functions

$$\beta \triangleq \prod_{i=1}^{M} \beta_i \tag{7}$$

which is similar to $\psi_\cup$ since it attains zero on the union of the zero level sets of the β_i's. Although the zero level set of β is typically larger than the boundary of the set described by $\psi_\cup(\beta_1, \cdots \beta_m) \leq 0$, it is simpler and will be adequate in many situations.

2) Scalar "Conditioning" Functions: Two real-valued functions of a single real argument will be used constantly in the sequel for the purposes of "conditioning" their argument without changing certain essential properties. In both cases, there is a scalar parameter that must be chosen to complete the definition, and we will emphasize the importance of this parameter by use of subscripts.

[13] Of course, a given set can be represented by many such obstacle functions.

The first function is a diffeomorphism from the extended reals $[0, \infty]$ onto the unit interval

$$\sigma_\lambda(x) \triangleq \frac{x}{\lambda + x} \tag{8}$$

where $\lambda > 0$ is a parameter. This function takes zero to zero, maps "the point at infinity" to unity, and varies smoothly in between. Its chief use in the sequel is to bound functions that achieve their maximal value uniformly on some set of interest by blowing up to $+\infty$ there. For example, an important use of this function is in the *analytic switch,* defined by

$$s(q, \lambda) \triangleq \left(\sigma_\lambda \circ \frac{\gamma}{\beta}\right)(q) = \frac{\gamma(q)}{\lambda\beta(q) + \gamma(q)}$$

(γ and β are implicit representations of the geometric data as described above). Note that this function vanishes exactly at the zeros of γ, achieves its upper bound of unity exactly at the zeroes of β, and varies smoothly between the two elsewhere.

The second function is a κth root function

$$\rho_\kappa(x) \triangleq x^{1/\kappa} \tag{9}$$

that we shall use to "sharpen" its argument. For example, $\gamma_2(q) = \|q - q_d\|^4$ is not a Morse function (since its Hessian at the origin is the zero matrix) while $\rho_2 \circ \gamma_2 = \|q - q_d\|^2$ is.

As an example of the use of these preliminary constructions, consider the navigation problem on the trivial Euclidean sphere world introduced in Section II-B-1. This problem may be solved with a sharpened analytic switch. Specifically, we have shown in [25, Theorem 1] that

$$\varphi_\kappa(q) \triangleq \left(\rho_\kappa \circ \sigma_1 \circ \frac{\gamma_\kappa}{\beta}\right)(q) = \frac{\|q - q_d\|^2}{[\|q - q_d\|^\kappa + \beta(q)]^{1/\kappa}} \tag{10}$$

(γ_κ is the distance to the goal, and β is the product of the spherical obstacle functions) is a navigation function, so long as the parameter κ exceeds a certain function of the geometric data (which is given in closed form in [25]). Fig. 3 shows a planar Euclidean sphere world with five internal obstacles. The destination is specified arbitrarily at the origin, and the level lines shown vary regularly between zero (at q_d) and unity (on the boundary). It can be seen that all the undesired local minima disappear as the parameter κ increases. The resulting navigation function has a unique minimum at q_d and a saddle point near each internal obstacle. As asserted in Section II-A, this is the best that can be achieved subject to the sphere world's topology.

3) Star-Shaped Sets and their Transformations: Fig. 4 depicts two sets that are both topological discs. $\mathcal{D}$ is a standard Euclidean disk with center p_i and radius ρ. $\mathcal{S}$ is a rather general shape called a "star shape." Star-shaped sets such as $\mathcal{S}$ are more general than (they include all) the convex sets and are characterized by possession of a distinguished "center point," $q_i \in \mathcal{S}$, from which all the rays cross their boundary once and only once.[14]

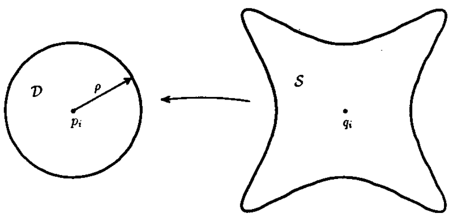


Fig. 4. A change of coordinates transforms the star-shaped set into a disk.

We now present a very useful function that has the following two properties: 1) it maps the boundary of the star diffeomorphically onto the boundary of the disk (i.e., to the sphere); 2) it maps the star's interior to the disk's interior and the star's "outside," $E^n - \mathcal{S}$, to the disk's outside. We shall assume that an implicit representation β_i has been given for $\mathcal{S}$, and we denote the implicit representation of the disk $\mathcal{D}$ by $\hat{\beta}_i$. The star-to-sphere transformation, which we call a *translated scaling* map, is defined by

$$T_i(q) \triangleq \nu_i \cdot [q - q_i] + p_i \tag{12}$$

(q, q_i, p_i are vectors, ν_i is a real-valued function of q). This transformation first scales each ray staring at q_i by the amount ν_i and then translates it along the vector p_i (the center of $\mathcal{D}$). The scaling factor ν_i is defined to be

$$\nu_i(q) \triangleq [1 + \beta_i(q)]^{1/2} \frac{\rho}{\|q - q_i\|} \tag{13}$$

(ρ is the disk's radius) and can be easily checked to satisfy the two desired properties. Thus, T_i maps the boundary of the star into the boundary of disk, its interior into the disk's interior, and its outside into the disk's outside as required.

III. Construction of Navigation Functions

This section, comprising the central contribution of the paper, exhibits families of coordinate transformations for increasingly complicated spaces in the class of the generalized sphere worlds.[15] Section II-A-4 above established that navigation functions on geometrically simple spaces induce navigation functions on all the topologically equivalent spaces via a diffeomorphism. Section II-B-2 presented a navigation function for the geometrically trivial Euclidean sphere world. Thus, the construction of transformations mapping geometrically complicated sphere worlds to the model sphere world solves the robot navigation problem posed in Section I-A.

In Section III-A, we introduce the underlying method of construction based upon the catalog of Section II-B and explain how it provides a solution in the case of a freespace whose obstacles are all star shaped. Next, in Section III-B we introduce the notion of "forests of stars" that greatly

[14] It was shown in [35] that the implicit representation of $\mathcal{S}$, β_i, typically satisfies

$$\nabla\beta_i(q) \cdot (q - q_i) \geq \Delta \quad \text{for all} \quad q \in \text{boundary } (\mathcal{S}) \tag{11}$$

for some constant $\Delta > 0$, assumed to be furnished along with the geometric data. This condition is needed in the proof of correctness, and its necessity is currently not clear.

[15] It has been suggested to us that certain classical techniques for building conformal mappings might be used in place of the original methods reported here. Indeed, a number of researchers have begun to use such constructions in a heuristic fashion [2], [20] for building artificial potential fields for robot motion control. We are not sure to what extent these methods of complex variable theory may overlap with our work.

increases the geometric detail we can represent. Finally, in Section III-C, we extend and generalize the construction of Section III-A yielding a recipe for diffeomorphisms between the forests of stars and the Euclidean sphere worlds. Each of the classes mentioned—Euclidean sphere worlds, star worlds, and forests of stars—is increasingly more "expressive" and is defined for any dimension of the Euclidean space and for arbitrary number and arrangement of the (disjoint) obstacles. The last class—the forests of stars—turns out to be capable of approximating any generalized sphere world.

A. Navigation Functions on Star Worlds

While a Euclidean sphere world is an n-dimensional disk punctured by an arbitrary number of smaller disjoint disks representing model obstacles, in a star world the obstacles are allowed to be n-dimensional star-shaped sets. These have already been described in Section II-B. Although they are topologically equivalent to the standard Euclidean disk as the translated scaling T_i demonstrates constructively, they are geometrically much more "expressive." For example, every convex set is star shaped (although many nonconvex sets are also star shaped).

In any event, the results of Section II-A now enable us to restate the problem of Section I-A as follows: Given a star world $\mathcal{F}$ with a goal q_d in its interior, find a suitable model sphere world $\mathcal{M}$ and construct a change of coordinates h between $\mathcal{F}$ and $\mathcal{M}$ such that $h(q_d) = p_d$, where p_d is an arbitrarily specified destination in the interior of $\mathcal{M}$.

1) Specification of the Model Sphere World: From a specification of the star world $\mathcal{F}$ we would like to construct a transformation mapping it to a model Euclidean sphere world $\mathcal{M}$. The first step is to determine a suitable $\mathcal{M}$.

$\mathcal{M}$ must have the same dimension and the same number of obstacles as $\mathcal{F}$, but there seems to be no obvious guideline for choosing the location and radii of the Euclidean spheres. We have found, however, that the transformation (defined below) becomes especially simple if the spheres are placed such that their centers coincide with the stars' centers (see Fig. 5). Further, we have found that the formula for choosing the parameter of the transformation becomes simpler if the radii of the internal spheres are chosen sufficiently small so that the Euclidean spheres are completely contained in the respective starts. Similarly, the outer sphere is chosen sufficiently large so that it contains the outer boundary of the star world (see Fig. 5). Finally, we have found it convenient to set the goal point in $\mathcal{M}$, p_d, to be identical to the desired goal in $\mathcal{F}$, q_d.

Note that this choice of $\mathcal{M}$ requires knowledge of a lower bound on the minimal distance of each star's center from its boundary (upper bound for the outer star). This geometric data is discussed in Appendix III.

2) Linear Combination of Translated Scalings: Each translated scaling T_i introduced in Section III-B-3 maps the boundary of one star to one simple Euclidean sphere. If there are many different boundaries that must be brought into correspondence with appropriately sized and located Euclidean spheres, then a single T_i will not do. Instead, we will introduce a "linear combination" of such maps whose relative weights change as a function of the configuration space.

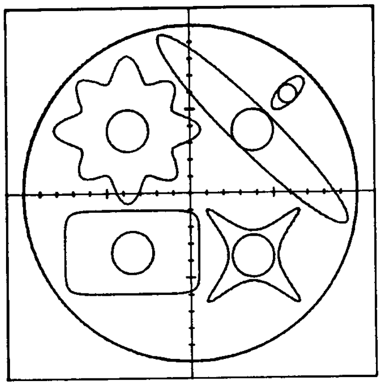


Fig. 5. A suitable model Euclidean sphere world superimposed on the given star world.

Consider a collection of $M+1$ disjoint star shapes, $\mathcal{S}_i$ for $i = 0, \ldots, M$, each with its corresponding translated scaling T_i, defined in terms of a scaling factor ν_i and implicit boundary representation β_i as in Section II-B-3 above. For each shape we form an analytic switch

$$s_i(q,\lambda) \triangleq \left(\sigma_\lambda \circ \frac{\gamma\overline{\beta}_i}{\beta_i}\right)(q) = \left(\frac{\gamma\overline{\beta}_i}{\gamma\overline{\beta}_i + \lambda\beta_i}\right)(q) \qquad (14)$$

where γ is the distance-to-the goal function (see (6)), and $\overline{\beta}_i$ is the "omitted product"

$$\overline{\beta}_i \triangleq \prod_{j=0, j\neq i}^{M} \beta_j.$$

This switch takes values in the unit interval, varyies between unity on the boundary of $\mathcal{S}_i$, and is zero on the boundary of every other shape, $\mathcal{S}_j$ for $j \neq i$, as well as at the destination (since γ appears as a factor in s_i). For the sake of notational consistency, we will denote the identify map on E^n as $T_d(q) = q$, and let

$$s_d(q,\lambda) \triangleq 1 - \sum_{i=0}^{M} s_i$$

denote the "destination switch."

Definition 2: Linear Combination of Translated Scalings is the one-parameter family of transformations defined by

$$h_\lambda(q) \triangleq s_d T_d(q) + \sum_{i=0}^{M} s_i T_i(q) \qquad (15)$$

(the parameter λ appears in the switches). Such functions resemble the translated scaling T_i in a neighborhood of the ith star boundary and match T_i exactly on the boundary. They resemble the identity map on configurations away from the boundaries and match the identity exactly at the destination. The rate of decrease of the switch s_i away from the boundary of the ith obstacle is governed by λ, and, as it turns out, a suitable choice of λ guarantees that h_λ is a diffeomorphism. The particular definitions of $\overline{\beta}_i$ and ν_i will vary with the geometric details of each case, but the form of the transformation will always be the same.

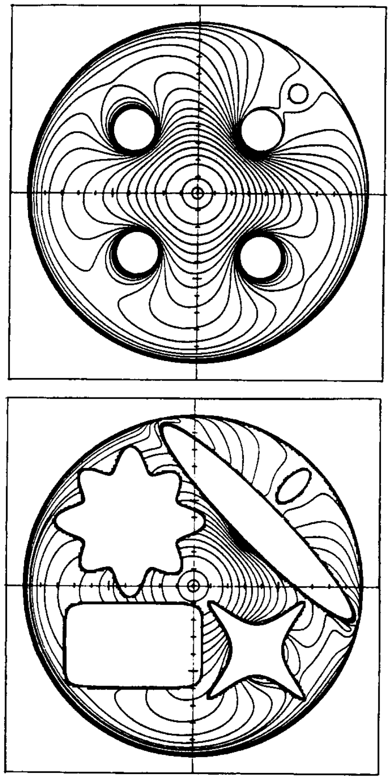


Fig. 6. A planar star world with five internal obstacles (bottom) and its model sphere world (top).

3) Linear Combination of Translated Scalings Solves the Star World Problem: We have shown in [38, theorem 1] that when the model space $\mathcal{M}$ is constructed as in Section III-A-1, then the linear combination (15) defines a diffeomorphism from $\mathcal{F}$ onto $\mathcal{M}$ so long as λ is chosen larger than some lower bound Λ. The proof of correctness actually constructs a formula for Λ in terms of q_d and the geometric data, listed in Appendix III. Thus, if $\hat{\varphi}$ is a navigation function for $\mathcal{M}$ and h_λ is the linear combination of translated scalings (15) with $\lambda \geq \Lambda$, then $\varphi \triangleq \hat{\varphi} \circ h_\lambda$ is a navigation function on the star world $\mathcal{F}$.

A numerical example of such an induced navigation function is provided in Fig. 6. The figure depicts a planar star world $\mathcal{F}$ with five internal obstacles and its corresponding model sphere world $\mathcal{M}$. The destination point in both spaces is chosen arbitrarily at the origin. The level lines of φ on $\mathcal{F}$ and of $\hat{\varphi}$ on $\mathcal{M}$ vary between zero (at q_d) and unity (on the boundary). The parameter in $\hat{\varphi}$ is chosen sufficiently high to eliminate spurious local minima in $\mathcal{M}$. It can be seen that for an appropriately chosen λ the resulting navigation function $\varphi = \hat{\varphi} \circ h_\lambda$ introduces no additional critical points. Thus, there is a unique minimum at q_d and one saddle point near each (internal) star obstacle. It was shown in Section II-A that one cannot do better than this with the sphere worlds' topology.

B. Extension to Geometrically Complicated Spaces

This section addresses the question of how to represent more geometrically complicated instances of the generalized sphere worlds than specified by the star worlds. We have chosen a representation obtained from decomposing configuration-space obstacles into unions of overlapping stars. This introduces a technical difficulty not encountered before: the obstacles' surfaces may now have kinks, or sharp corners, at the "seams" where two stars intersect each other. We shall first formulate the notion of an obstacle with sharp corners in Section III-B-1. Next, we shall introduce the "forest of stars" in Section III-B-2 as a concrete instance of how allowing such unions of overlapping shapes can yield a representation of obstacles that is essentially complete.

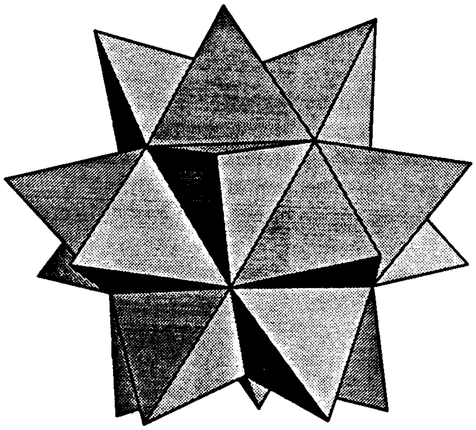


Fig. 7. Every polyhedron is a semianalytic set.

1) Semianalytic Obstacles: A set $\mathcal{S} \subset E^n$ is *semianalytic* [34] if it can be expressed as a finite Boolean combination (via the set operations $\cup$, $\cap$, $-$, c) of sets $\mathcal{S}_i$, each described by a real-valued function β_i in the form

$$\mathcal{S}_i \triangleq \{q \in E^n : \beta_i(q) \leq 0\}. \tag{16}$$

For example, any polyhedron is a finite intersection of linear inequalities (see Fig. 7). We shall allow only nondegenerate semianalytic shapes. Namely, n-dimensional sets whose boundary is a disjoint union of $(n-1)$-dimensional smooth *patches* (or faces), "glued" together along their respective boundaries (or edges). We denote by $\mathcal{C}$ the collection of these edges, and call this set the *sharp corners* of $\mathcal{S}$. For example, the planes forming the boundary of the polygon in Fig. 7 satisfy this property and, in fact, almost all semianalytic shapes satisfy this property [14, pp. 69–69]. We call such sets *semianalytic obstacles.*

2) Forest of Stars: The notion of semianalytic set may be used to greatly increase the geometric expressiveness of our representation of obstacles. We may now consider unions of overlapping stars. We would like, however, to allow only unions that are generalized n-dimensional disks. For this purpose, we restrict the stars' adjacency graph to be a tree (i.e., a connected graph having no edge loops) as follows. Describe the arrangement of the stars comprising each obstacle by an adjacency graph whose vertices are the star centers and whose edges connect centers of overlapping stars. Thus, a *tree-of-stars* is a finite union of overlapping stars whose adjacency graph is a tree. A *forest of stars* is an n-dimensional star-

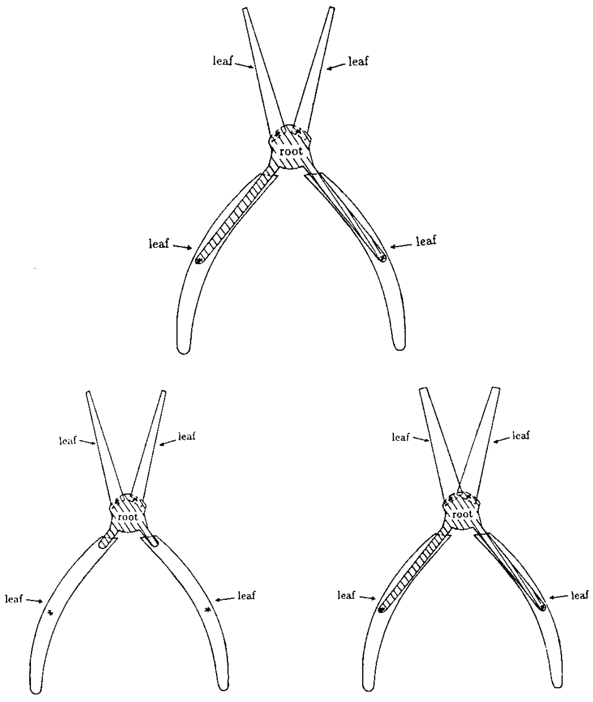


Fig. 8. A simple tree of stars (top) and two prohibited situations (bottom).

shaped set, denoted by $\mathcal{T}_0$,[16] punctured by an arbitrary number of smaller disjoint tree-of-stars *obstacles,* denoted by $\mathcal{T}_i$ for $i = 1, \ldots, M$. In the special case in which each tree of stars consists of one star, the resulting space is a star world defined in Section III-A.

We shall find it convenient to restrict attention to a simplified subclass of the forests of stars whose members are guaranteed to be generalized sphere worlds. The first simplification is to restrict the shapes to be *quadratic.* Namely, the various stars must be expressed as a Boolean combination of polynomial inequalities of degree at most two. In other words, the boundary of each star is partitioned into planar and quadratic patches with sharp corners allowed. The second simplification requires that each tree of stars $\mathcal{T}_i$ satisfies the following three properties with respect to some choice of its root:[17] 1) the center point of each star is contained in its parent in the tree; 2) each of the stars is connected to its parent via a unique patch; 3) this patch is star shaped with respect to the center point of the star it connects to (i.e., the rays from this center intersect the patch at most once). We shall call such forests of stars *simple.* Fig. 8 shows an example of a valid tree of stars as well as two prohibited situations. On the left, not all the centers of the leaves are inside their parent. On the right, two leaves intersect each other so that the stars' intersection arrangement is not a tree.

The convenience afforded by these restrictions does not come at the expense of any generality since we have shown that any generalized sphere world can be approximated by such forests of stars [35].[18] Moreover, if all the stars are also convex, then the forest of stars is guaranteed to be simple. From now on, the term *forest of stars* will mean simple and quadratic as well. Finally, we have relegated to Appendix I a rather technical discussion concerning the extension of the class of navigation functions to spaces with piecewise smooth obstacle surfaces, such as the forest of stars.

C. Navigation Functions on Forest of Stars

We now address the central problem of the paper. Namely, the construction of a coordinate transformation mapping a given forest of stars to its model sphere world. We shall do it by successively "purging" the forest with the following basic operation: Given a forest of stars $\mathcal{F}$ with maximal tree depth $d \geq 1$, and with a destination q_d in its interior, find a model forest of stars $\hat{\mathcal{F}}$ with maximal tree depth $d - 1$. Construct a change of coordinates f_λ between $\mathcal{F}$ and $\hat{\mathcal{F}}$ such that $p_d = f_\lambda(q_d)$, where p_d is an arbitrarily specified destination point in the interior of $\mathcal{F}$.

Notice that a forest of stars of depth zero is a star world. Thus, a composition of such transformations yields a diffeomorphism between the original forest and a star world. The latter may then be treated by the methods of Section III-A. This section introduces the "purging transformation" that maps $\mathcal{F}$ onto $\hat{\mathcal{F}}$. Section IV describes several simulation studies, indicating the rich variety of shapes attainable with forests of starts and depicting problems that may arise with practical implementation.

1) The Purging Transformation: Given a forest of stars $\mathcal{F}$, we label its stars by considering a tree at a time: for each tree, designate one of its stars as the root and then label the stars by proceeding along edge paths. The order by which the stars along a specific path are encountered determines a parent–child relation: we denote the index of the parent of the ith star by $p(i)$. Each such path ends in a leaf. We denote the index set of all the leaves in the forest that belong to trees consisting of more than one star by $\mathcal{L}$, and denote the index set of all the stars in $\mathcal{F}$ by $\mathcal{I}$. We are now ready to present the purging transformation. As before, it is given in terms of the distance-to-the-goal function γ and the implicit representation of the stars' boundary β_i for $i \in \mathcal{I}$.

Definition 3: The *purging transformation* f_λ is a linear combination of translated scalings T_i that range over the leaves $i \in \mathcal{L}$

$$f_\lambda \triangleq s_d T_d(q) + \sum_{i\in\mathcal{L}} s_i T_i(q)$$

where $s_d = 1 - \sum_{i\in\mathcal{L}} s_i$, and the switches $s_i(q, \lambda)$ have the same structure as before, except the omitted products appearing in the switches, which now become

$$\overline{\beta}_i \triangleq \left(\prod_{k\in\mathcal{I}-\{i,p(i)\}} \beta_k\right)\cdot\left(\prod_{k\in\mathcal{L}-\{i\}} \beta_k\right)\cdot\tilde{\beta}_{p(i)}. \qquad (17)$$

[16] $\mathcal{T}_0$ can be an "inverted" tree-of-stars, one whose root is the complement of a star-shaped set $\mathcal{S}$ and whose other stars "grow" into $\mathcal{S}$. For simplicity, assume that $\mathcal{T}_0$ is star shaped.

[17] Recall that any node of a tree can be designated as its root.

[18] The approximation is with respect to the *Hausdorff metric,* defined on the class of all compact subsets of E^n [44]. Unfortunately, and characteristically of this metric, an approximating obstacle does not necessarily contain the original obstacle, thus limiting its practical use.

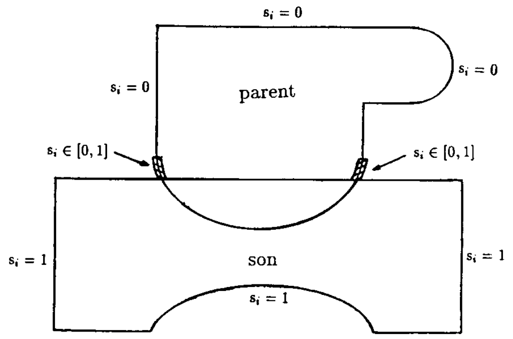


Fig. 9. The various values of the switch s_i.

It is worth contrasting the intuition underlying this construction with that of the simpler star-world to sphere-world transformation of Section III-A. There are two essential complications. First, the switches s_i must be modified to account for the new situation that several different translated scalings T_i must operate on different portions—the distinct leaves—of the same connected boundary—the ith tree of stars. Second, the translated scalings T_i are now designed to bring the exposed portion of a leaf's boundary into correspondence with the portion of the parent's boundary that it encloses rather than a simple sphere.

As shown in Fig. 9, each s_i attains the value of unity on the ith leaf's boundary, becomes zero on the boundary of every other star except its parent, and varies smoothly between the two in the interior of $\mathcal{F}$. Note that whenever $s_i = 0$ for some $i \in \mathcal{L}$, the transformation f_λ becomes the identity map. Since we would like f_λ to be the identity map on the boundary of the parent stars, it is desirable that s_i be zero on the boundary of the ith parent. This parent, however, is connected to the ith leaf on whose boundary the switch s_i is identically unity. This apparent contradiction is resolved by distinguishing a "collar" portion within the parent's boundary in a small neighborhood about the ith leaf via the special function $\tilde{\beta}_{p(i)}$ (note, according to the definition of a simple tree in Section III-B-2, that no other leaf intersects a sufficiently small collar).

Specifically, $\tilde{\beta}_{p(i)}$ is the implicit representation of the region obtained by intersecting the ith parent and the complement of a "thickened" version of its child (the maximal amount of thickening is governed by a geometric constant that is part of the geometric data). The desired $\tilde{\beta}_{p(i)}$ is easily constructed using Zenkin's formulas and is given in Appendix II. Thus, $\tilde{\beta}_{p(i)}$ effectively represents a new "obstacle" formed by removing from the parent a thickened version of its child. Since $\tilde{\beta}_{p(i)}$ is zero exactly on the boundary of this region and is positive elsewhere in $\mathcal{F}$ (and, in particular, on the ith collar), the switch s_i of (17) becomes zero on the boundary of the ith parent excluding the collar and attains the value unity on the ith leaf. On the collar itself, which lies within the parent's boundary, s_i varies smoothly between unity on the outer rim and zero on the inner rim—the "seam" between the leaf and its parent (see Fig. 9).

It should now be clear that the purging transformation exactly matches T_i on the boundary of each leaf, exactly matches the identity transformation on the boundary of all the nonleaf stars (excluding a collar around each parent-leaf seam) and at the goal point q_d, and varies smoothly between zero and unity on the interior of $\mathcal{F}$. The behavior of f_λ on the collars will become clear from a discussion of the component transformations T_i.

The ith translated scaling T_i maps the ith leaf's boundary onto a portion of its parent's boundary. Namely, T_i scales the rays starting at the leaf's center point q_i and crossing the leaf's boundary at points q by the scaling factor $\nu_i(q)$.[19] On the collar surrounding the ith parent-leaf seam, the switch s_i varies between zero and unity. Consequently, f_λ varies between the identity map on the collar's outer rim and the translated scaling T_i on the inner rim, which is the seam itself. Inspection of ν_i shows that T_i matches the identity map on the collar. Thus, f_λ reduces to the identity transformation over the entire collar region, and consequently it is the identity transformation over the entire boundary of the parent, as desired.

2) The Purging Transformation Yields a Navigation Function: We conclude this section with a statement of the theorem asserting the correctness of f_λ. First there is a technical detail to be clarified. In general, the smoothness of the navigation function is sufficient to guarantee that the resulting feedback control law (2) will be bounded, as required by the practical torque limits of the robot's actuators. Yet the construction presented here is merely continuous at the obstacles' sharp corners $\mathcal{C}$ as revealed by inspecting the scaling factors ν_i and the functions $\tilde{\beta}_{p(i)}$ (see Appendix II). We present here this less desirable construction since it is simpler than its smooth version. For this situation, however, one must explicitly verify that the Jacobian of f_λ is bounded on $\mathcal{F}$ because this would imply the boundedness of the gradient of the resulting navigation function φ. This fact is, therefore, explicitly stated in the following theorem:

Theorem 1 (from [37]): For any (simple and quadratic) forest of stars $\mathcal{F}$, there exists a positive constant $\Lambda > 0$ such that if $\lambda \geq \Lambda$ then the purging transformation f_λ is a one-to-one function that maps $\mathcal{F}$ onto its purged version $\hat{\mathcal{F}}$ and is a *diffeomorphism* away from the obstacles' sharp corners $\mathcal{C}$. Moreover, the Jacobian matrix of f_λ is bounded on $\mathcal{F} - \mathcal{C}$, provided that the gradient of the obstacle functions $\nabla\beta_i$ for $i \in \mathcal{I}$ is bounded on $\mathcal{F} - \mathcal{C}$.

The proof of correctness actually constructs a formula for Λ in terms of q_d and the geometric data (see Appendix III). To summarize, the mapping of the forest of stars $\mathcal{F}$ onto its model star world is achieved by successively purging the trees in $\mathcal{F}$. Each purging stage yields a forest of smaller depth, and this process ends when each tree has only its root left, which is a star. Since it was already shown how to construct a navigation function on star worlds, the problem of constructing a navigation function on $\mathcal{F}$ is solved. This is illustrated in the next section.

[19] The conceptual distinctions between the sphere-based scaling ν_i (see (13)) and the more general version required here are slight enough that the technically involved redefinition is relegated to Appendix II.

IV. Implementation

This section discusses implementation issues concerning the navigation functions on the forest of stars: their computational complexity and several observations drawn from their numerical simulation.

A. Computational Complexity

The computation involved has two parts. First, when presented with the geometric data describing the forest $\mathcal{F}$, a navigation function is constructed on $\mathcal{F}$ according to the recipe

$$\varphi = \hat{\varphi} \circ h_\lambda \circ f_{\lambda_d} \circ \cdots f_{\lambda_1}$$

where $\hat{\varphi}$ is a navigation function on the corresponding Euclidean sphere world $\mathcal{M}$, h_λ maps the star world whose obstacles are the roots of the trees in $\mathcal{F}$ onto $\mathcal{M}$, and f_{λ_i}, for $i = 1, \cdots, d$ are successive purging transformations. The construction of φ involves the choice of $d+2$ parameters for the $d+2$ functions comprising φ. The computational cost of this part is shown in [35] to be bounded by

$$10M^2n + 15M^2 + 25d|\mathcal{I}| + |\mathcal{I}|n^3$$

where M is the number of tree-like obstacles, $|\mathcal{I}|$ the total number of star obstacles, d the depth of the forest, and n the dimension of the ambient Euclidean space. It is important to note that this count does not include the computation of the various data terms (see Appendix III). In the simulations described below, we directly specify the obstacle functions and adjust the $d+2$ parameters to be "sufficiently high" without ever computing the various geometric constants.

Second, the controller has to compute $\nabla\varphi$

$$\nabla\varphi = \nabla(\hat{\varphi} \circ h_\lambda \circ f_{\lambda_d} \circ \cdots f_{\lambda_1}). \tag{18}$$

It is shown in [35] that this computation is bounded by

$$5d|\mathcal{I}| + 15|\mathcal{I}|n^2 + 5dn^3 + 5d\sum_{j\in\mathcal{I}}(\#\beta_j + \#\nabla\beta_j)$$

where $\#(\beta_j)$ and $\#(\nabla\beta_j)$ are "place holders" for the computational cost of β_i and $\nabla\beta_i$. These terms are not completely unknown: each obstacle function is obtained from a Boolean combination of m linear and quadratic inequalities. It is shown in [35] that these can be effectively arranged in a recursive form so that their computation is *linear* in m.

As a concrete time measure, the time to evaluate $\nabla\varphi$ along the gradient trajectories depicted on the trivial example of Fig. 10 (that involves one coordinate transformation) is 4 ms on a DEC5000 machine. Roughly speaking, the time required for more complicated examples grows linearly in the number of stars comprising the tree obstacles $|\mathcal{I}|$ and linearly in the number of coordinate transformations d required to obtain the corresponding model Euclidean sphere world.

B. Simulation Studies

Fig. 11 depicts a forest of stars resembling a building floor plan. There are three internal tree-like obstacles, and the depth of the deepest tree is $d = 4$. The purging transformation f_{λ_i} is applied d times until a space whose obstacles are the roots of the original trees is obtained. This space is a star world, and the previously constructed star-world to sphere-world transformation is applied to obtain the corresponding model sphere world $\mathcal{M}$. Thus, the sequence of transformations is

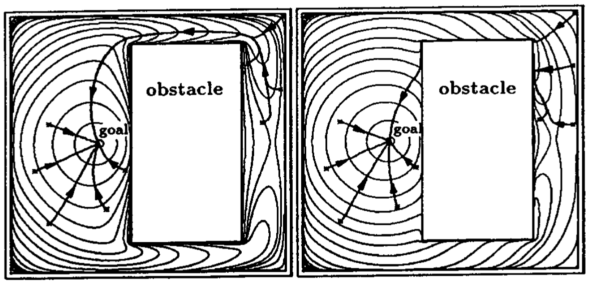


Fig. 10. The phenomenon of disappearing valleys. The valleys are clearly depicted for low parameter values (left) but disappear as the parameter values increase (right).

$$\mathcal{F} \xrightarrow{f_{\lambda_1}} \mathcal{F}_1 \xrightarrow{f_{\lambda_2}} \mathcal{F}_2 \xrightarrow{f_{\lambda_3}} \mathcal{F}_3 \xrightarrow{f_{\lambda_4}} \mathcal{F}_4 \xrightarrow{h_\lambda} \mathcal{M}.$$

The figure depicts each of the intermediate spaces as well as the level lines of the navigation function as it is "pulled back" via these spaces. The destination point is chosen arbitrarily at the origin, and the level lines shown vary between zero (at q_d) and unity (on the boundary). The parameter κ of the original navigation function on $\mathcal{M}$ is chosen sufficiently high to eliminate spurious local minima, according to [25, Theorem 1]. The parameters in the transformations f_λ and f_{λ_i} for $i = 1, \cdots, 4$ are chosen sufficiently high so that no new local minima appear, according to [38, Theorem 1] and Theorem 1 here. Thus, there is a unique minimum at q_d and one saddle point near each (internal) tree-like obstacle. As we have shown in Section II-A, one cannot do better than this using smooth vector fields.

The simulations, while corroborating the theory, reveal certain numerical difficulties. The level lines clearly depict a unique minimum at q_d. But, when the various "valleys" approach the obstacles, they are so close to the boundary that they can be seen again only on the far side of each obstacle, as portrayed in Fig. 10. We have found from further numerical experimentation that such situations—when the valleys hug the boundaries too tightly—lead to gradient vector fields that vary too abruptly to be implemented in a practical setting. The phenomenon becomes even more acute when the parameters' values increases. Of course, the theory requires in general that these parameters be increased to avoid spurious local minima.

To understand this behavior, consider the purging transformation

$$\vec{f}_\lambda(q) = \sum_{i\in\mathcal{L}} s_i(q,\lambda)\vec{f}_i(q) + \left(1 - \sum_{i\in\mathcal{L}} s_i(q,\lambda)\right)\vec{\mathrm{id}}(q).$$

The effect of increasing the parameter λ is to make the switches s_i vanish more rapidly away from the obstacles' boundary, rendering the transformation essentially the identity mapping. Thus, all the "interesting" features are confined to small neighborhoods about the obstacles' boundaries.

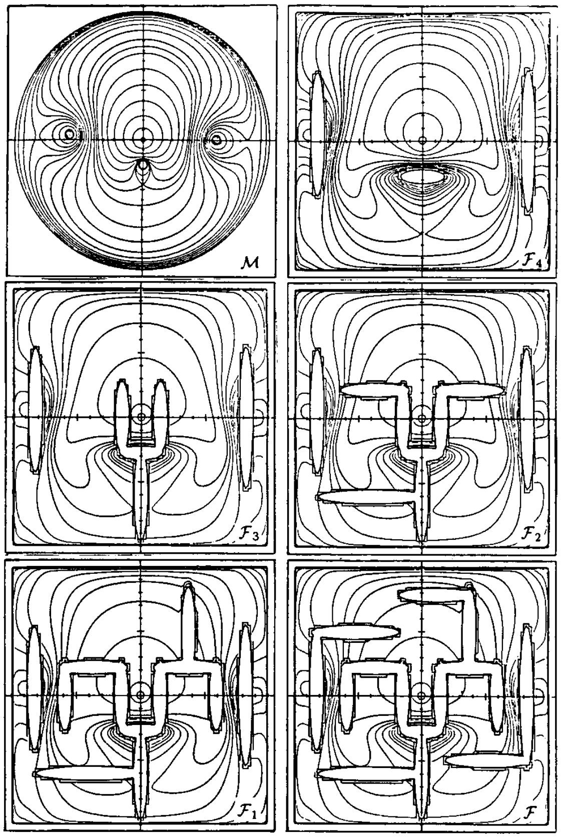


Fig. 11. Planar forest of stars with three internal tree-like obstacles (bottom right), its purged versions, and its model sphere world (top left).

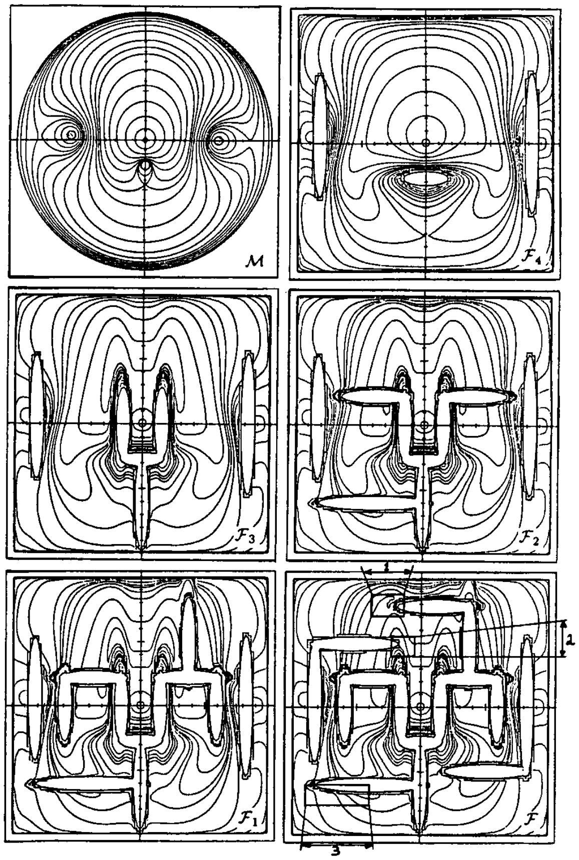


Fig. 12. Result of experimentation on the space of Fig. 10.

In Fig. 12 we have experimented with various intuitive numerical remedies for the problem of disappearing valleys. Additional parameters introduced in the switches s_i make their value close to unity in a neighborhood about the boundaries and only then decrease to zero. Intuitively, this slows the transition of the purging transformation to the identity map, and consequently the valleys move away from the obstacles' boundary. Fig. 13 depicts the enlarged image of some interesting regions in $\mathcal{F}$, demonstrating the "reappearance" of the valleys.

It becomes clear from these simulations that the practicability of this construction depends on the development of some control mechanism for the location of the valleys. This is the subject of research now in progress.

V. Conclusion

This paper motivates and defines formally *navigation functions,* a new class of artificial potential functions that simultaneously solve the two principal aspects of the robot motion planning and control problems. First, navigation functions are correct with respect to the path-planning problem. Namely, almost all the negative-gradient trajectories converge to the desired destination q_d, and none can leave the free space $\mathcal{F}$. In general, global convergence with smooth vector fields is precluded by the topology of $\mathcal{F}$. Second, navigation functions provide automatically a controller that implements the path correctly. Namely, their gradient vector field gives rise to a bounded-torque feedback control law that guarantees that the closed-loop robot system will approach q_d without hitting obstacles. Moreover, there is an explicitly specified set of initial positions and velocities, including any initial position on the boundary of $\mathcal{F}$ with zero velocity, for which the robot trajectories exhibit the desired behavior.

The chief contribution is a detailed "recipe" for constructing navigation functions on almost any generalized n-dimensional sphere world whose geometry is *a priori* known. This results in closed-form expressions that depend upon q_d, the geometric data, as well as certain free parameters that affect the shape of the potential function. Lower bounds on the parameters are computed in terms of the geometric data, guaranteeing that, for suitable values, the constructions are indeed navigation functions.

These results provide for the first time a smooth bounded potential-function-based robot navigation algorithm that is provably correct. But, although smooth navigation functions always exist and their underlying control theory is completely general, the generalized sphere worlds treated here constitute a very small portion of the general case of an n-link robot navigating between arbitrarily shaped obstacles. Some impor-

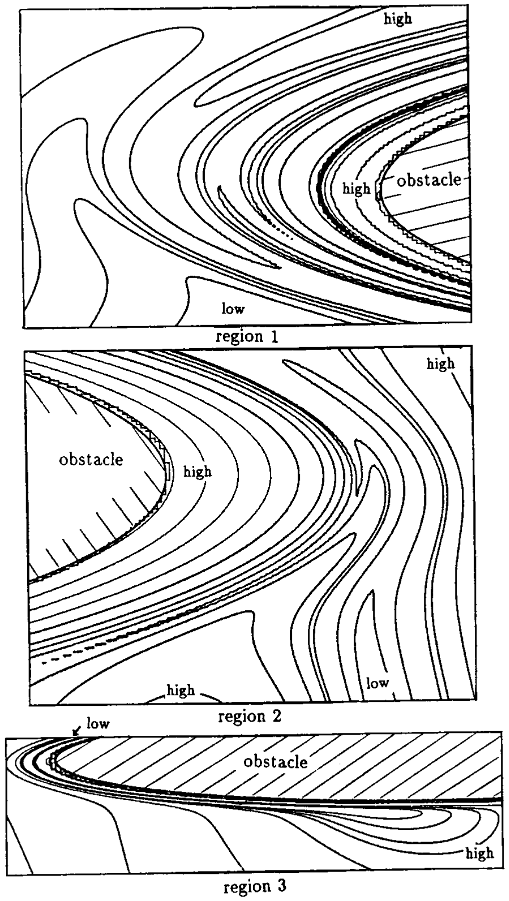


Fig. 13. The enlarged image of some regions.

tant situations can be modeled in practice with this machinery. For example, it can be shown that the free configuration space of a disk-shaped robot moving in the plane amidst arbitrary shapes is a collection of generalized sphere worlds (excluding pathological cases). Our methods provide a means of building up a solution for progressively more accurate representations of such sphere worlds. Yet, ideally, one would like to construct navigation functions for progressively more realistic robotic situations in general.

Under the naive assumptions of Section I, a program of research aimed at this goal should take the following hierarchy of difficulty into consideration: 1) a point-mass robot navigating amidst arbitrary shapes; 2) a rigid body navigating in the same space; 3) an n-link robot in the same space; 4) several robots moving in a coordinated motion amidst obstacles. In fact, there has been some progress since these results were obtained in pursuing such problems [36], [23]. However, our naive assumptions—stationary obstacles, fixed destination, perfect information, ideal sensors, and ideal bounded-torque actuators—are themselves unrealistic, and their relaxation is imperative in the long run. The questions raised by these possibilities are discussed in brief below.

Consider first the matter of perfect information. The most computationally problematic aspect of this requirement arises from the present need to compute obstacle functions as closed-form symbolic expressions, extracting information about the connectivity of the resulting free space, and computing the constants that determine the navigation function parameters along the way. Beyond the present activity in generating implicit representations for obstacles there is some hope that "adaptive" versions of our ideas might be pursued successfully. For example, it would be preferable (both computationally and philosophically) to leave the ultimate choice of the lower bound on the critical free parameters that enter our constructions to the wisdom of an automatic adjustment rule that raises or lowers their magnitude until the destination is reached. Ongoing work in this area suggest that some provably correct means of doing so might be forthcoming. More radically, it might be possible to replace known obstacle functions themselves with known parameterized families whose specific parameter values are adjusted as progressively more about the geometric details of the configuration space is learned in the course of navigation. Eventually, a proper reliance upon the distance function as measured by a suitably clever sensory device might substitute in large measure for computation. This last prospect of gathering information effectively about a stationary world "on the fly" raises the even more attractive possibility of treating worlds in motion as well. However, all of these remarks represent mere speculation at the present time.

Appendix I
An Extended Definition of the Navigation Functions

The class of navigation functions for configuration spaces with semianalytic obstacles must be extended from the following reason. Consider a free configuration space $\mathcal{F}$ resulting from the removal of semianalytic obstacles from the configuration space. According to the admissibility property, the boundary of $\mathcal{F}$ must be a level set of the navigation function φ. But, the boundary of a semianalytic obstacle is nondifferentiable since its tangent space at the corner points is not well defined. Thus, on the set of sharp corners φ must be either nondifferentiable or have a vanishing gradient. The latter option must be adopted since, in general, φ must be at least of class $C^{(2)}$ (twice continuously differentiable) to guarantee the classical existence and uniqueness results for the trajectories of the closed-loop robot system. It follows that the gradient of φ must vanish at the corner points, making the corners degenerate local maxima. This, in turn, implies that φ cannot be a Morse function on the entirety of $\mathcal{F}$, as summarized in the following definition:

Definition 4 (Definition 1 Revised): Let $\mathcal{F}$ be a free configuration space resulting from the removal of semianalytic obstacles, and let $\mathcal{C}$ be the obstacles' sharp corners. A map $\varphi : \mathcal{F} \rightarrow [0, 1]$ is a *navigation function* if it satisfies Definition 1 with the exception that now φ is required to be Morse only away from the sharp corners on, the set $\mathcal{F} - \mathcal{C}$.

The fact that φ is not required to be Morse on the entirety of $\mathcal{F}$ requires a reconsideration of the underlying control theory. One must guarantee that the resulting feedback control law still guides the physical robot correctly. Although this is beyond

$$\nu_i(q) \triangleq \begin{cases} \frac{-\pi(q_i)+\kappa_i(q)}{\delta_i(q)+\kappa_i(q)} & \text{if } \mathcal{P}_i \text{ is planar} \\ \frac{(q-q_i)^T Q_i(p_i-q_i)+\sqrt{\kappa_i(q)}+\operatorname{sgn}\left(D^2\pi_i\right)\sqrt{\delta_i(q)+\kappa_i(q)}}{(q-q_i)^T Q(q-q_i)} & \text{if } \mathcal{P}_i \text{ is quadratic} \end{cases} \tag{B2}$$

the scope of this paper, we believe that a simple energy-conservation argument ensures the physical viability of this extended class.

The invariance of navigation properties under coordinate transformation must also be restated for the following reason. It was previously noted that navigation functions are invariant under diffeomorphic transformations. Yet, such transformations cannot be constructed on a space whose boundary has kinks. Instead, the transformation must be allowed to have a singular Jacobian at the sharp corners. The extended invariance property can be stated as follows. Let $\mathcal{M}$ and $\mathcal{F}$ be two free configuration spaces. Let $\varphi : \mathcal{M} \longrightarrow [0,1]$ be a navigation function on $\mathcal{M}$, and let the smooth map $f : \mathcal{F} \longrightarrow \mathcal{M}$ be both one-to-one and onto, and be a diffeomorphism away from $\mathcal{C}$; then $\tilde{\varphi} = \varphi \circ h$ is a navigation function on $\mathcal{F}$.

Appendix II
Some Details Concerning the Forest of Stars

This appendix describes terms that appear in the definition of the purging transformation f_λ, whose definition was omitted in order to simplify the presentation.

Let us first describe the function $\tilde{\beta}_{p(i)}$ appearing in the switches. Using Zenkin's formula [46], this function is defined by

$$\tilde{\beta}_{p(i)}(q) \triangleq \beta_{p(i)}(q) + (2E_i - \beta_i(q)) + \sqrt{\beta^2_{p(i)}(q) + (2E_i - \beta_i(q))^2} \tag{B1}$$

where E_i is a geometric constant whose exact nature is discussed in [35]. The function $\tilde{\beta}_{p(i)}$ corresponds to a new "obstacle," obtained by excluding from the ith parent (described by $\beta_{p(i)}$) a $2E_i$-thickened version of the ith leaf (described by $\beta_{p(i)}$, as illustrated in Fig. 14. By construction, the interior of the parent is disjoint from $\mathcal{F}$, and since $\tilde{\beta}_{p(i)}$ is negative only inside the parent, we have that $\tilde{\beta}_{p(i)} \geq 0$ on $\mathcal{F}$. On the boundary of the parent we have that $\beta_{p(i)} = 0$. Consequently, $\tilde{\beta}_{p(i)}$ is strictly positive on the "collar" region surrounding the ith leaf and contained in the boundary of its parent.

The other term in the purging transformation whose definition was omitted is the ith star-set scaling factor ν_i, appearing in the translated scaling map T_i. Consider the ith leaf for some $i \in \mathcal{L}$. By construction it is "connected" to its parent via a unique patch. This patch, denoted by P_i, is contained in the zero level set of a known polynomial:

$$\pi_i(q) \triangleq \begin{cases} (q-p_i)\cdot v_i - c_i, & \text{if } \mathcal{P}_i \text{ is planar} \\ (q-p_i)^T Q_i (q-p_i) - 1 & \text{if } \mathcal{P}_i \text{ is quadratic} \end{cases}$$

where v_i is a nonzero n vector, and Q_i is an $n \times n$ positive definite (symmetric) matrix.

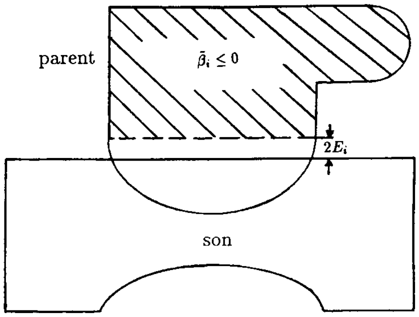


Fig. 14. The set described by $\tilde{\beta}_{p(i)}$.

Remark: This paper does not treat the whole list of quadratic shapes. Augmenting the construction presented below with other quadratic shapes—for instance, cylinders and cones—requires more tedious algebra but resembles the ellipsoidal and planar shapes discussed here.

The *star-set deforming factors* are the real-valued functions defined on $\mathcal{F}$ by (B2), which is given at the top of this page. As we have already said, the purpose of $\nu_i(q)$ is to scale the rays starting at the leaf's center point q_i such that points q on the leaf's boundary are mapped to points where the rays cross the patch. See [35] for details concerning the other terms in (B2).

Appendix III
The Required Geometric Data

In this appendix we describe the geometric data required in the construction of the navigation functions. We shall describe the data required for the *forests of stars,* since this class includes the other ones—the Euclidean sphere worlds and the star worlds—as special subclasses.

The required data can be divided into two groups: The first is symbolic in nature and describes the geometry of the free configuration space. It includes the adjacency graph of the stars in each tree-of-stars obstacle, the obstacle functions for the various stars, and the stars' centers. The second is a list of geometric constants used in the determination of theoretically correct parameter values for the navigation functions.[20] In principle, these constants can be extracted from the obstacle functions, but we have not developed any general procedures to compute them. In practice, we have obtained navigation functions simply by increasing the various parameters to "sufficiently high" values. Moreover, since the

[20] An exception is the constant E_i appearing in the construction itself (see (B1)). It is currently not clear whether the presence of this constant in the construction is actually necessary.

obstacle functions in the forests of stars are constructed from planar and quadratic polynomials, explicit formulas giving bounds (exact values are never required) on these features ought to be much easier to obtain.

The list of these constants is rather lengthy, so let us first describe the constants required for computing the parameter of the navigation function on the *star worlds*. These constants are a subset of those required for the forests of stars. First, for each star obstacle, a lower bound on its minimal "radius"—the minimal distance from its center q_i to its boundary (upper bound on the radius of the zeroth obstacle). Second, upper bounds on the obstacle functions and on the distance-to-the-goal function γ, denoted by $\{E_{i1}\}_0^M$ and E_d, guaranteeing that the "thickened boundaries" of any two stars still do not overlap

$$\{q : \beta_i(q) \leq E_{i1}\} \bigcap \{q : \beta_j(q) \leq E_{j1}\} = \emptyset \quad \text{for } i \neq j. \tag{C1}$$

Nor do they overlap an E_d disk about the destination

$$\{q : \beta_i(q) \leq E_{i1}\} \bigcap \{q : \gamma(q) \leq E_d\} = \emptyset.$$

Third, for each obstacle function β_i, a pair of positive constants (Δ_i, E_{i2}) satisfying the inner-product condition of (11)

$$\nabla\beta_i(q) \cdot (q - q_i) \geq \Delta_i \qquad \forall\, q \in \mathcal{F} \quad \text{such that } \beta_i(q) \leq E_{i2}. \tag{C2}$$

Last, we require for each obstacle function β_i knowledge of the following upper bounds:

$$\max_{\mathcal{F}} \{\beta_i\} \quad \text{and} \quad \max_{\mathcal{F}} \{\|\nabla\beta_i\|\}.$$

Additionally, there is a list of three constants required exclusively for the forest of stars, for instance, E_i used to construct the parent-leaf collars in the switches (see (B1)). The details of these constants are rather involved and can be found in [35].

References

[1] M. D. Adams, H. Hu, and P. J. Roberts, "Towards a real-time architecture for obstacle avoidance and path planning in mobile robots," in *Proc. IEEE Int. Conf. Robotic Automat.* (Cincinnati, OH), May 1990, pp. 584–589.

[2] S. Akishita, S. Kawamura, and K.-I. Hayashi, "New navigation function utilizing hydrodynamic potential for mobile robot," in *Proc. IEEE Motion Control Conf.*, 1990.

[3] J. R. Andrews and N. Hogan, "Impedance control as a framework for implementing obstacle avoidance in a manipulator," in *Control of Manufacturing and Robotic Systems*, D. E. Hardt and W. J. Book, Eds. Boston: ASME, 1983, pp. 243–251.

[4] J. Barraquand, B. Langlois, and J. C. Latombe, "Robot motion planning with many degrees of freedom and dynamic constraints," in *Proc. 5th Int. Symp. Robotics Res.* (Tokyo), Aug. 1989, pp. 74–83.

[5] J. Barraquand and J. C. Latombe, "A Monte-Carlo algorithm for path-planning with many degrees of freedom," in *Proc. IEEE Int. Conf. Robotics Automat.* (Cincinnati, OH), May 1990, pp. 1712–1717.

[6] N. P. Bhatia and G. P. Szego, *Dynamical Systems: Stability Theory and Applicaitons.* New York: Springer-Verlag, 1967.

[7] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic maniupulators along specified paths," *Int. J. Robotics Res.*, vol. 4, no. 3, pp. 3–17, Fall 1985.

[8] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst. Man Cybern.*, vol. 19, no. 5, pp. 1179–1187, 1989.

[9] J. F. Canny, "The complexity of robot motion planning," Ph.D. dissertation, Dept. Electrical Eng. Computer Sci., Massachusetts Inst. Technol., Cambridge, MA, May 1987.

[10] C. I. Connolly, J. B. Burns, and R. Weiss, "Path planning using Laplace's equation," in *Proc. IEEE Int. Conf. Robotics Automat.* (Cincinnati, OH), May 1990, pp. 2102–2106.

[11] B. Donald and P. Xavier, "Near-optimal kinodynamic planning for robots with coupled dynamics bounds," in *Proc. IEEE Int. Symp. Intell. Contr.* (Albany, NY), Sept. 1989, pp. 354–359.

[12] B. R. Donald, "Error detection and recovery for robot motion planning with uncertainty," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, 1987.

[13] E. Freund, "Fast nonlinear control with arbitrary pole placement for industrial robots and manipulators," *Int. J. Robotics Res.*, vol. 1, no. 1, pp. 65–78, 1983.

[14] V. Guillemin and A. Pollack, *Differential Topology.* Englewood Cliffs, NJ: Prentice-Hall, 1974.

[15] J. E. Hopcroft and G. Wilfong, "Motion of objects in contact," *Int. J. Robotics Res.*, vol. 4, no. 4, pp. 32–46, Winter 1986.

[16] O. Khatib, "Commande dynamique dans l'espace opérational des robots manipulateurs en présence d'obstacles," Ph.D. dissertation, École nationale Supéieure de l'Aeéronatique et de l'Espace (ENSAE), France, 1980.

[17] ——, "Real time obstacle avoidance for maniupulators and mobile robots," *Int. J. Robotics Res.*, vol. 5, no. 1, pp. 90–99, Spring 1986.

[18] O Khatib and J.-F. Le Maitre, "Dynamic control of manipulators operating in a complex environment," in *Proc. Int. CISM-IFToMM Symp.* (Udine, Italy), Sept. 1978, pp. 267–282.

[19] P. Khosla and R. Volpe, "Superquadric artifical potentials for obstacle avoidance and approach," in *Proc. IEEE Int. Conf. Robotics and Automat.* (Philadelphia, PA), Apr. 1988, pp. 1778–1784.

[20] J.-O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," Tech. Rep., Carnegie-Mellon Univ., Pittsburgh, PA, 1990.

[21] D. E. Koditschek, "Natural motion for robot arms," in *Proc. 23rd IEEE Conf. Decision Contr.* (Las Vegas, NV), Dec. 1984, pp. 733–735.

[22] ——, "Globally stable closed loops imply autonomous behavior," in *Proc. 5th IEEE Int. Symp. Intell. Contr.* (Philadelphia, PA), Sept. 1990.

[23] ——, "An approach to autonomous robot assembly," *Robotica*, 1991, submitted for publication.

[24] ——, "The control of natural motion in mechanical systems," *ASME J. Dynam., Syst., Meas. Contr.*, Sept. 1991.

[25] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances Appl. Math.*, vol. 11, pp. 412–442, 1990.

[26] B. H. Krogh, "A generalized potential field approach to obstacle avoidance control," in *Proc. SME Conf. Robotics Res.* (Bethlehem, PA), Aug. 1984.

[27] T. Lozano-Perez and al. Handey, "A robot system that recognizes, plans, and manipulates," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1987, pp. 843–849.

[28] T. Lozano-Perez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *Int. J. Robotics Res.*, vol. 3, no. 1, pp. 3–23, 1984.

[29] J. Y. S. Luh, M. W. Walker, and R. P. Paul, "Resolved acceleration control of mechanical manipulators," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 468–474, 1980.

[30] F. Miyazaki and S. Arimoto, "Sensory feedback based on the artificial potential for robot manipulators," in *Proc. 9th IFAC* (Budapest, Hungary), July 1984.

[31] W. S. Newman, "High-speed robot control in complex environments," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, 1987.

[32] W. S. Newman and N. Hogan, "High speed robot control and obstacle avoidance using dynamic potential functions," in *Proc. IEEE Int. Conf. Robotics Automat.* (Raleigh, NC), 1987, pp. 14–24.

[33] V. V. Pavlov and A. N. Voronin, "The method of potential functions for coding constraints of the external space in an intelligent mobile robot," *Sov. Automat. Contr.*, vol. 6, 1984.

[34] A. A. G. Requicha and H. B. Voelcker, "Solid modeling: Current status and research directions," *IEEE Comput. Graphics Applicat.*, vol. 3, no. 10, pp. 25–37, Oct. 1983.

[35] E. Rimon, "Exact robot navigation using artificial potential functions," Ph.D. dissertation, Yale Univ., New Haven, CT, 1990.

[36] ——, "A navigation function for a simple rigid body," in *Proc. IEEE Int. Conf. Robotics Automat.* (Sacramento, CA), Apr. 1991.

[37] E. Rimon and D. E. Koditschek, "Exact robot navigation in geometrically complicated but topologically simple spaces," in *Proc. IEEE Int. Conf. Robotic Automat.* (Cincinnati, OH), May 1990, pp. 1937–1943.

[38] ——, "The construction of analytic diffeomorphisms for exact robot navigation on star worlds," *Trans. Amer. Math. Soc.,* vol. 327, no. 1, pp. 71–116, Sept. 1991.

[39] J. T. Schwartz and M. Sharir, "On the "piano movers" problem. II. General techniques for computing topological properties of real algebraic manifolds," *Advances Appl. Math.,* vol. 4, no. 1, pp. 298–351, 1983.

[40] S. Smale, "Generalized Poincaré's conjecture in dimensions greater than four," *Ann. Math.,* vol. 74, no. 2, pp. 391–406, 1961.

[41] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," vol. 102, pp. 119–125, 1981.

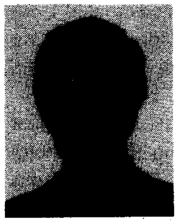
[42] W. Thompson and P. G. Tait, *Treatise on Natural Philosophy.* Cambridge, U.K.: Cambridge, 1886.

[43] R. B. Tilove, "Local obstacle avoidance for mobile robots based on the method of artificial potentials," in *Proc. IEEE Int. Conf. Robotics Automat.* (Cincinnati, OH), May 1990, pp. 566–571.

[44] F. A. Toranzos, "Approximation of compact star-shaped sets by special families," *Rev. Unión Mathemática Argentia,* vol. 29, pp. 49–54, 1979.

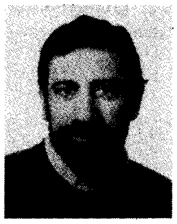
[45] C. W. Warren, "Global path planning using artificial potential fields," in *Proc. IEEE Int. Conf. Robotics Automat.* (Scottsdale, AZ), May 1989, pp. 316–321.

[46] O. V. Zenkin, "On the analytic description of geometric shapes," *Kibernetika,* vol. 4, pp. 103–110, 1970 (in Russian).



Elon Rimon (S'89–M'91) received the B.Sc. degree in 1985 in computer engineering from the Technion, Israel and the Ph.D. degree in 1990 from Yale University, New Haven, CT.

He was a Post-Doctoral Fellow at Stanford University during 1991. He is presently a Lecturer at the Electrical Engineering Department of the University of California at Berkeley. His interests include geometrical resoning in robotics, application of robot motion-planning techniques to computer graphics, and real-time sensory-based robot control.



Daniel E. Koditschek (S'80–M'83) received the B.S. degree in 1977 in engineering and applied science and the Ph.D. degree in 1983 in electrical engineering, both from Yale University, New Haven, CT.

He was appointed Assistant Professor of Electrical Engineering at Yale in 1984 where he is presently an Associate Professor. His interests include the application of dynamical systems theory to autonomous machine design, noninear control theory, and the application of computational theory and hardware to feedback control of physical processes.