May 26, 1992

Applications of Harmonic Functions to Robotics

Christopher I. Connolly
Roderic Grupen, *University of Massachusetts - Amherst*

Applications of Harmonic Functions to Robotics

Christopher I. Connolly and Roderic A. Grupen

Laboratory for Perceptual Robotics,

Computer and Information Science Department,

University of Massachusetts at Amherst *

May 26, 1992

Abstract

Harmonic functions are solutions to Laplace's Equation. As noted in a previous paper, they can be used to advantage for potential-field path planning, since they do not exhibit spurious local minima. In this paper, harmonic functions are shown to have a number of other properties (including completeness) which are essential to robotics applications. These properties strongly recommend harmonic functions as a mechanism for robot control.

1 Introduction

Harmonic functions were proposed in Connolly et al. [1] as a means for generating paths in a robot configuration space. These functions are solutions to Laplace's Equation. One important feature of harmonic functions is that they can be used to generate smooth, collision-free paths without the threat of spurious local minima. In this paper, harmonic functions are shown to provide a global, complete path-planning scheme. As noted by Tarassenko and Blake [2], a resistive grid can compute such functions very quickly, further motivating their use.

It is important to note that nearly all the potential field models in use prior to [1] were not based on charge distribution. If they had been, then the resulting functions would have been harmonic, hence free of spurious local minima. The possible exception to this is the work of Rimon

---

and Koditschek [3], whose functions also appear to be free of spurious local minima. The main motivations for the introduction of harmonic functions in [1] were the elimination of local minima and the use of a true analogy to phenomena such as charge distribution and fluid flow. In fact, the mathematical literature often refers to Laplace's Equation as the "potential equation" (see, for example, Courant and Hilbert [4]).

In this paper, harmonic functions are shown to have several more useful properties which make them well suited for robotics applications:

- Fast surface normal computation

- Completeness (up to discretization error in the environment model)

- Ability to exhibit different modes of behavior (grazing vs. avoidance)

- Robust control in the presence of unanticipated obstacles and errors

2   Harmonic functions

A harmonic function on a domain $\Omega \subset \mathbf{R}^n$ is a function which satisfies Laplace's equation:

$$\nabla^2 \phi = \sum_{i=1}^{n} \frac{\partial^2 \phi}{\partial x_i^2} = 0$$

In the case of robot path construction, the boundary of Ω ($\partial \Omega$) consists of the boundaries of all obstacles and goals in a configuration space representation. Harmonic functions satisfy the min-max principle (see [4, 1]): Spontaneous creation of local mimima within the region is impossible if Laplace's equation is imposed as a constraint on the functions used. In the context of this paper, harmonic functions are computed over a regular grid of points which is a discrete representation of the robot configuration space. A finite difference scheme is often used to compute harmonic functions (see [1] for details), and one can also employ a resistive grid [2].

In [1], Dirichlet-style boundary conditions were used. With Dirichlet boundary conditions, obstacles are raised to a constant high potential, while goal regions are kept at a low potential. The resulting potential in free space is then constrained by $\nabla^2 \phi = 0$. When Dirichlet boundary conditions are used in this manner, the gradient of ϕ is aligned with the surface normals of the obstacles. If this function is used for path generation, it will tend to *repel* the robot away from obstacles.

In [2], a Neumann-style boundary condition is proposed. Here, rather than keep obstacles at a constant potential, the derivative of ϕ is constrained so that $n \cdot \nabla \phi = 0$, where n is the surface

normal function for the obstacles[1]. This causes $\nabla\phi$ to be tangential to obstacle surfaces. Here, the paths generated will *graze* any obstacles they encounter.

As noted in [1] and [2], the Dirichlet solution tends to have precision problems. Flat regions can develop resulting in very small (but of necessity nonzero) gradients, requiring higher precision in generating the solution. On the other hand, the grazing behavior exhibited by a Neumann solution can result in a tendency for a robot to stay close to obstacle surfaces.

3   Properties

The original motivation for the use of harmonic functions arose from the prevention of spurious local minima. However, there are several other useful properties than can be exploited. The following sections elaborate on these properties.

3.1   Surface Normals

In subsequent sections, some use will be made of the fact that the gradient of a constant-value Dirichlet solution to Laplace's Equation will be normal to the surfaces of boundaries. The Dirichlet solution for $\nabla^2\phi = 0$ (as described in section 2) causes any obstacle boundary to become an equipotential surface. Therefore, the gradient $\nabla\phi_D$ must be orthogonal to this surface [5]. Hence $\nabla\phi_D$ must be aligned with object surface normals. Several algorithms in computer vision rely on the computation of surface normals ([6], [7]). In combination with a resistive grid [2], harmonic functions provide a very fast method for surface normal computation on discrete solid models (e.g. bitmap or octree models of objects – see [8] for an example of how such data could be acquired).

3.2   Completeness

Under certain assumptions regarding the boundary conditions, it is possible to show that a path planning scheme using harmonic functions is complete (up to the approximation of the environment). We assume the following:

- Ω is a compact subset of $\mathbf{R}^n$.

- ϕ is a solution to Laplace's Equation on Ω with Dirichlet boundary conditions.

- Streamlines for ϕ are computed via gradient descent.

---

[1] In general, Neumann boundary conditions are described by $n \cdot \nabla\phi(\mathbf{x}) = f(\mathbf{x})$, where $\mathbf{x} \in \mathbf{R}^n$. To get the gradient to flow *around* obstacles, one can simply set $f \equiv 0$.

- All boundaries $\partial\Omega$ are held at the obstacle potential $\phi = 1$.

- All goal regions are held at the goal potential $\phi = 0$.

As noted previously, every streamline of a harmonic function must reach a goal point.

Conjecture 1 *Let $\mathbf{x} \in \mathbf{R}^n$ be the manipulator starting configuration, and $\mathbf{y} \in \mathbf{R}^n$ be any manipulator goal configuration. There is no path between $\mathbf{x}$ and $\mathbf{y}$ iff. $\nabla\phi(\mathbf{x}) = 0$.*

Proof: Let $S \subset \Omega$ denote the connected component of freespace which contains $\mathbf{x}$.

$\Rightarrow$ If there is no path from $\mathbf{x}$ to $\mathbf{y}$, it must be the case that $\mathbf{x} \in S$ and $\mathbf{y} \notin S$. Therefore, the only potentials within S must be the high potentials ($\phi = 1$) on ∂S. By the min-max property of harmonic functions, both the minimum and the maximum potentials in any neighborhood of $\mathbf{x}$, and since the neighborhood is arbitrary, any point in S, must be the same, i.e. $\phi(\mathbf{x}) = 1$. Therefore, ϕ is constant and $\nabla\phi$ at any point in S must be zero.

$\Leftarrow$ If $\nabla\phi(\mathbf{x}) = 0$ in S, then the derivatives of ϕ must be zero and ϕ must be a constant function. Consider a point $\mathbf{z} \in S - \partial S$. The min-max property requires that the minimum and maximum values of ϕ in any neighborhood of $\mathbf{z}$ must be attained on the boundary of the neighborhood. Let a given neighborhood of $\mathbf{z}$ contain points in ∂S. By the min-max property, since ϕ is constant, it must also be true that $\phi(\mathbf{z}) = 1$ (the value at ∂S). Since the neighborhood is arbitrary, $\phi(\mathbf{z}) = 1$ must hold for any $\mathbf{z} \in S$. For $\mathbf{y}$ to be in S then would be a violation of the min-max property, since $\phi(\mathbf{y}) = 0$. Hence $\mathbf{y}$ cannot be in S and there is no path between the effector and the goal.

Similar arguments may be applied to the case of Neumann boundary conditions. The key to these arguments is simply that the function in any connected component without goals will converge to a constant in that region (see Courant and Hilbert [4] for a detailed treatment of this property).

3.3 Behavior

Solutions to Laplace's Equation obey the principle of superposition. By superposing the Dirichlet and Neumann solutions, one can obtain a harmonic function that exhibits a behavior somewhere between the two original solutions. Let ϕ_D be a Dirichlet solution on a domain Ω, and let ϕ_N be a Neumann solution on Ω. A new harmonic function can be constructed by taking a linear combination of the two:

$$\phi = k\phi_D + (1 - k)\phi_N$$

where $k \in [0, 1]$. By varying k, the new function can be made to exhibit some mixture of the "repel" and "graze" behaviors inherent in the original solutions. It is easy to envision situations where either behavior might be more desirable than the other. If k is kept away from zero, the

precision problems inherent in the Dirichlet solution are avoided. The resulting ϕ is harmonic, has no local minima and, provided that $k \in [0, 1]$, guarantees collision-free paths.

To see why the collision-free property is maintained, consider what happens to harmonic functions at the boundaries of obstacles. The gradient of a Dirichlet solution is outward normal to the surface at these points, while the gradient of a Neumann solution is parallel to the surface. Any convex combination of the two solutions will therefore yield a gradient which is bounded by the Dirichlet and the Neumann gradient. This gradient must have a component which is outward normal to the surface.

The variability of robot behavior is especially useful in two ways: Repulsion from obstacles is desirable in fragile environments, where the environment model is accurate and where contact is to be avoided if at all possible. It is also desirable in the case where the robot system itself is fragile or can be damaged by contact with the environment. On the other hand, the grazing behavior of the Neumann solution is useful as an exploratory strategy, in cases where contact with obstacles is acceptable and even desirable. Force sensing can then be used to update the environment model.

To illustrate the effect of this style of superposition, consider the configuration space of a two-jointed robot where each link is 1 unit long. A rectangular obstacle is placed near the robot, and a smaller rectangular goal region is placed just above the obstacle (figure 1). Figures 2, 3 and 4 show the configuration space that results, and the streamlines obtained from the superposed harmonic functions for $k = 0$, $k = 1$, and $k = 0.5$ respectively. In each figure, the right side depicts the robot configuration space, while the left side shows its cartesian workspace. The lines in cartesian space show the paths of the end effector corresponding to the streamlines in configuration space. Lightly shaded regions are obstacles, and black regions are goal points.

Note that in figure 3, many paths graze the configuration space obstacles (the black region). In contrast, figure 2 shows streamlines which tend to stay well clear of obstacles. Finally, 4 shows the streamlines exhibiting a behavior somewhere between the two.

3.4 Velocity Control

Velocity control can be used to great advantage with harmonic functions. The gradient of a harmonic function can be used to supply velocity commands at every point in the robot workspace. In the absence of external forces, a robot so commanded will follow a streamline[2] in configuration space, which will always lead to a goal configuration [9]. Deflections in the streamline which result from external forces will simply place the robot on another streamline.

For example, admittance control [10, 11, 12] is ideally suited for use with harmonic functions.

---

[2]The integral curve resulting from the gradient descent of the harmonic function.
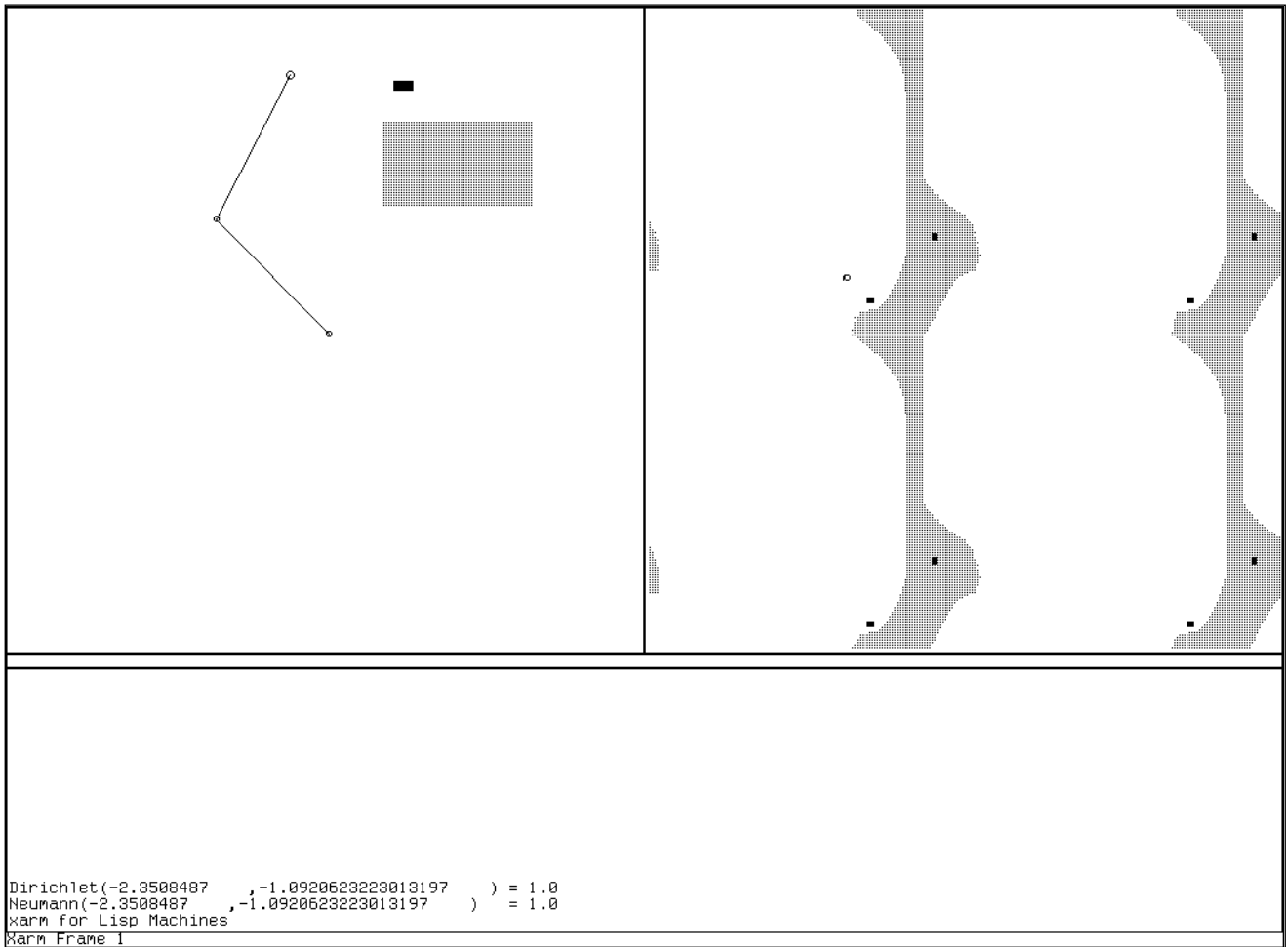
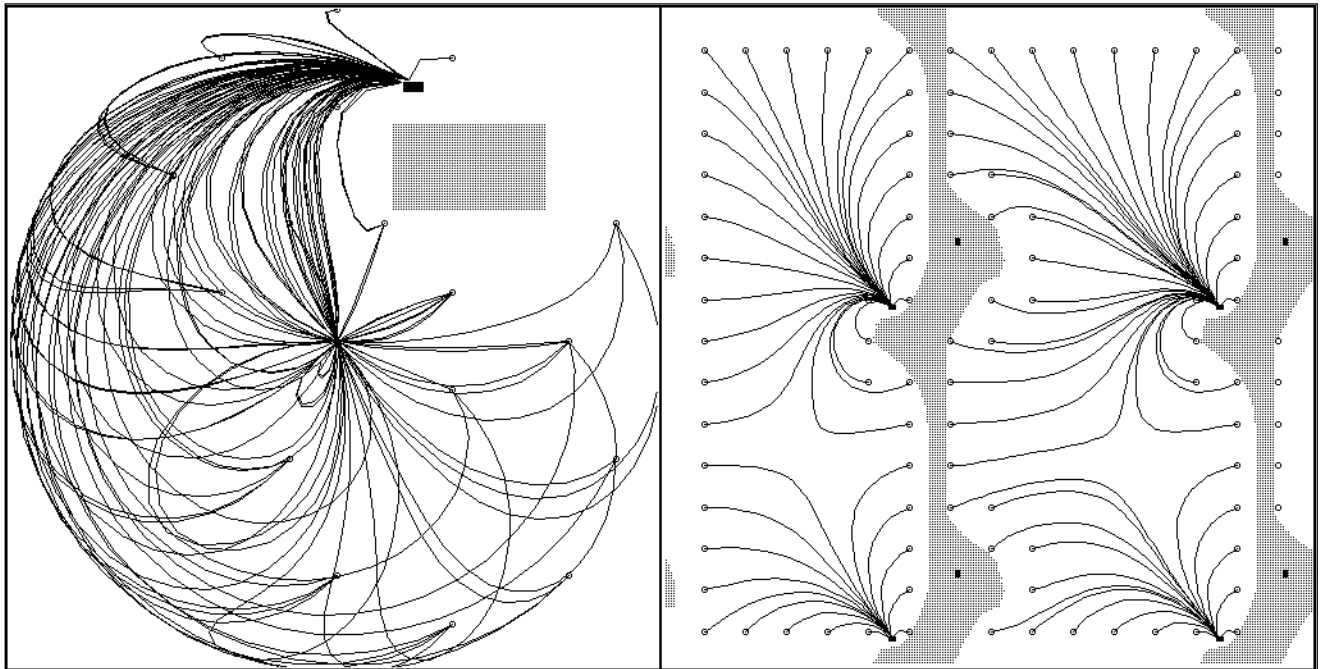Figure 1: Cartesian space (left) and configuration space (right) for Problem A.

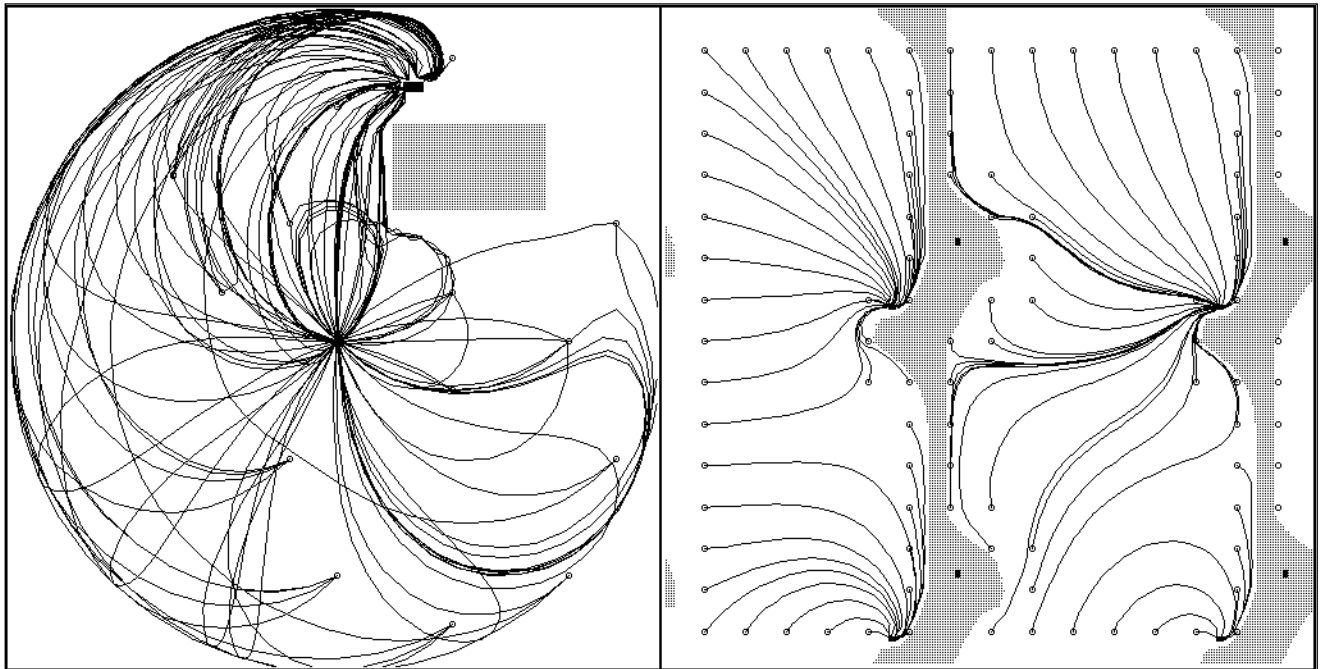Figure 2: Streamlines for the Dirichlet solution of Problem A.

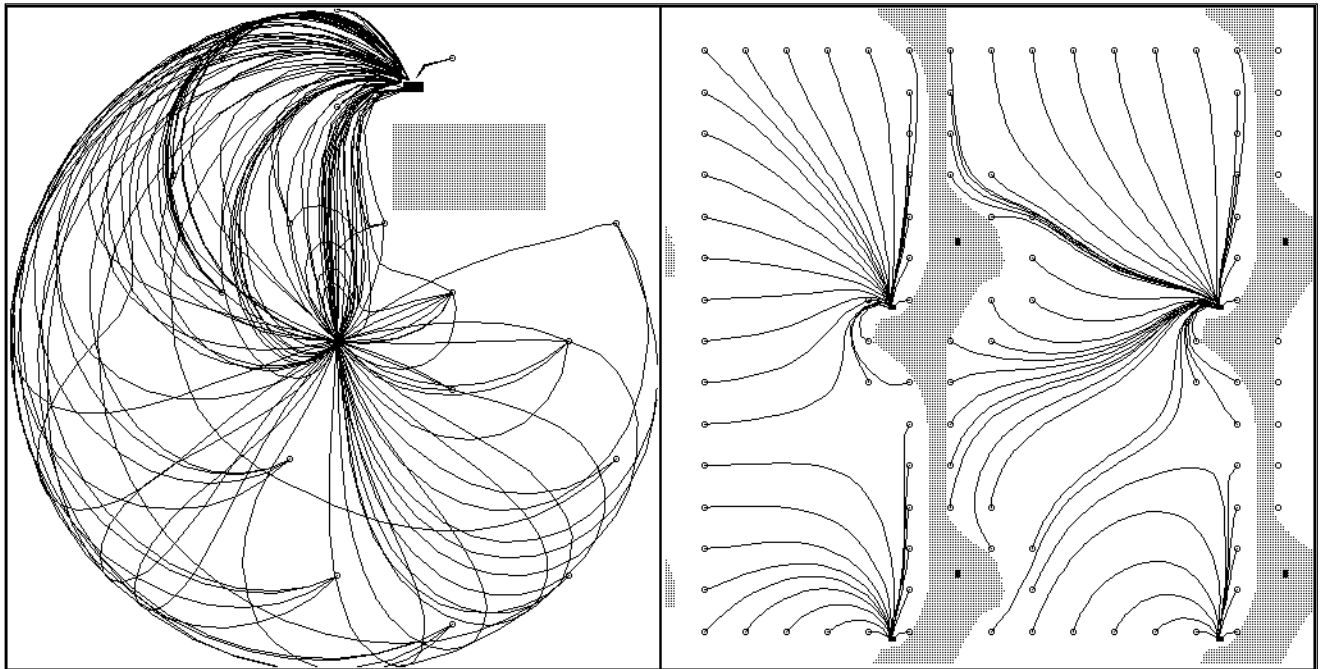Figure 3: Streamlines for the Neumann solution of Problem A.

Figure 4: Streamlines for Problem A with a superposition constant of 0.5.

This control law is as follows:

$$\mathbf{V} = \mathbf{V_0} + \mathcal{A}\mathbf{F}$$

where $\mathcal{A}$ is an admittance matrix that alters the nominal command velocities $\mathbf{V_0}$ in response to external forces and torques $\mathbf{F}$. The scaled gradient of a harmonic function can be used as the command velocity $\mathbf{V_0}$.

A robot which uses harmonic functions to provide velocity control in this manner is robust in the presence of uncertainty in the robot's model of its environment. Errors in the model of the environment can result in contact between the robot and obstacles. This causes the robot to be deflected from its path. Since every streamline of a harmonic function *must* terminate at a goal point, the system can simply sample the harmonic function gradient at the current robot configuration and follow it. The application of harmonic functions to velocity control is described in more detail in Section 5.

4 Examples

A system has been implemented which demonstrates the use of harmonic functions for control of a revolute robot arm. The system simulates a two-link planar robot which uses harmonic functions for generating paths. In the experiments shown here, the left side of each figure depicts the cartesian space grid, while the right side shows the corresponding configuration space grid. As before, lightly shaded regions denote obstacles, while solid black regions are goals. Each harmonic function is computed over the configuration space grid. Not shown is a "virtual" obstacle which in each case is placed at singular points of the manipulator Jacobian matrix (diagonal lines in cspace).

Figures 5, 6, and 7 all show paths generated from the same starting configuration, but with different superposition constants. The letter "S" denotes the start configuration. Note the tendency of the end effector to roam closer to the obstacles as the constant approaches zero.

In figures 8, 9, and 10, the ability of harmonic functions to use more than one goal region is demonstrated. These examples also help to show that harmonic functions are capable of dealing with a variety of environments.

Finally, figures 11, 12 and 13 show the ability of harmonic functions to detect that no path can be obtained from start to goal. This is a two-goal problem in which certain configurations are isolated from both goals. In this case, it is sufficient to test that the gradient is zero and that the harmonic function value is 1. Note that in figure 11, a path is found from start to goal. However, in figure 12, there is no path, and the program terminates and notifies the user that this is the case.
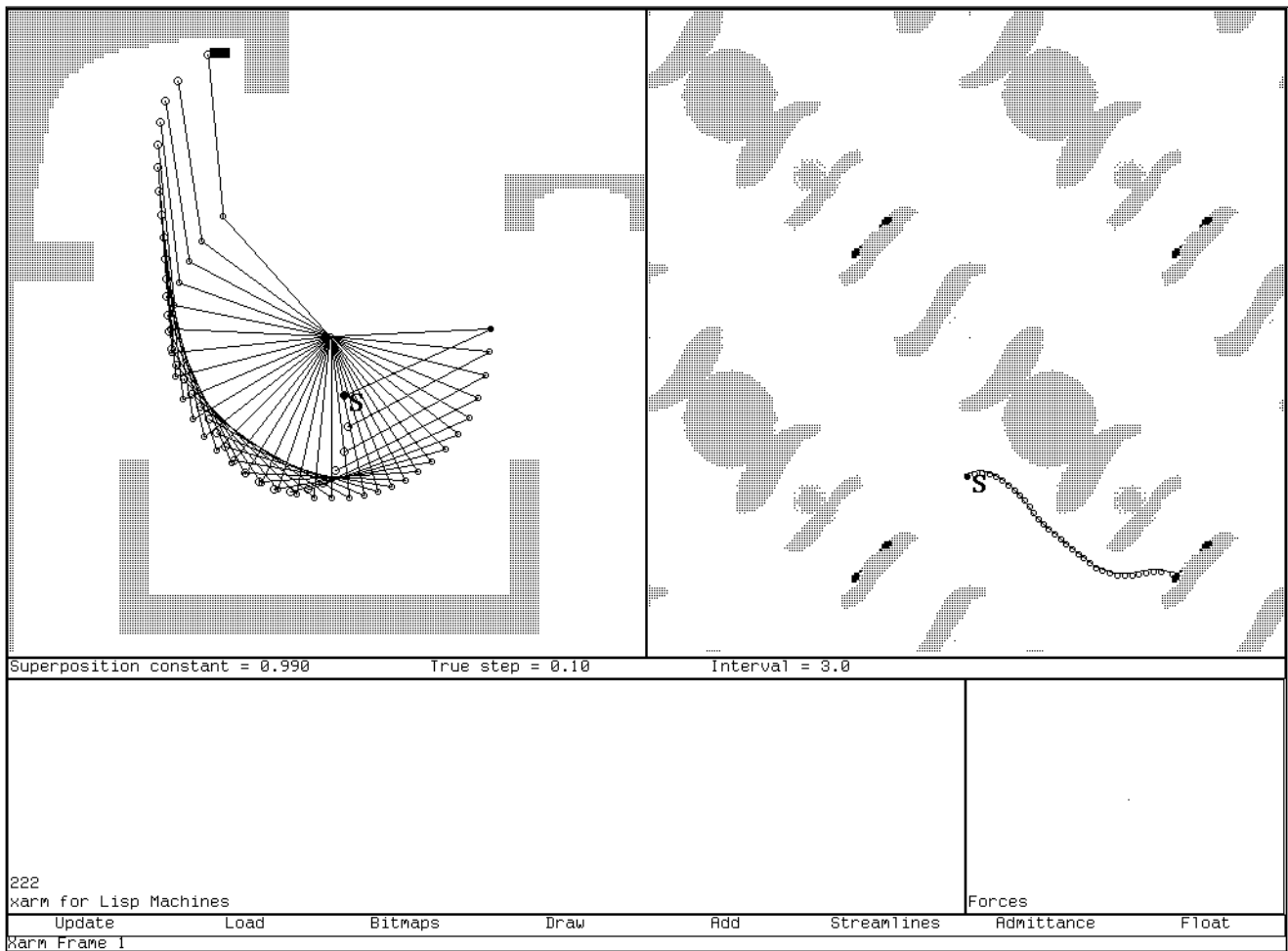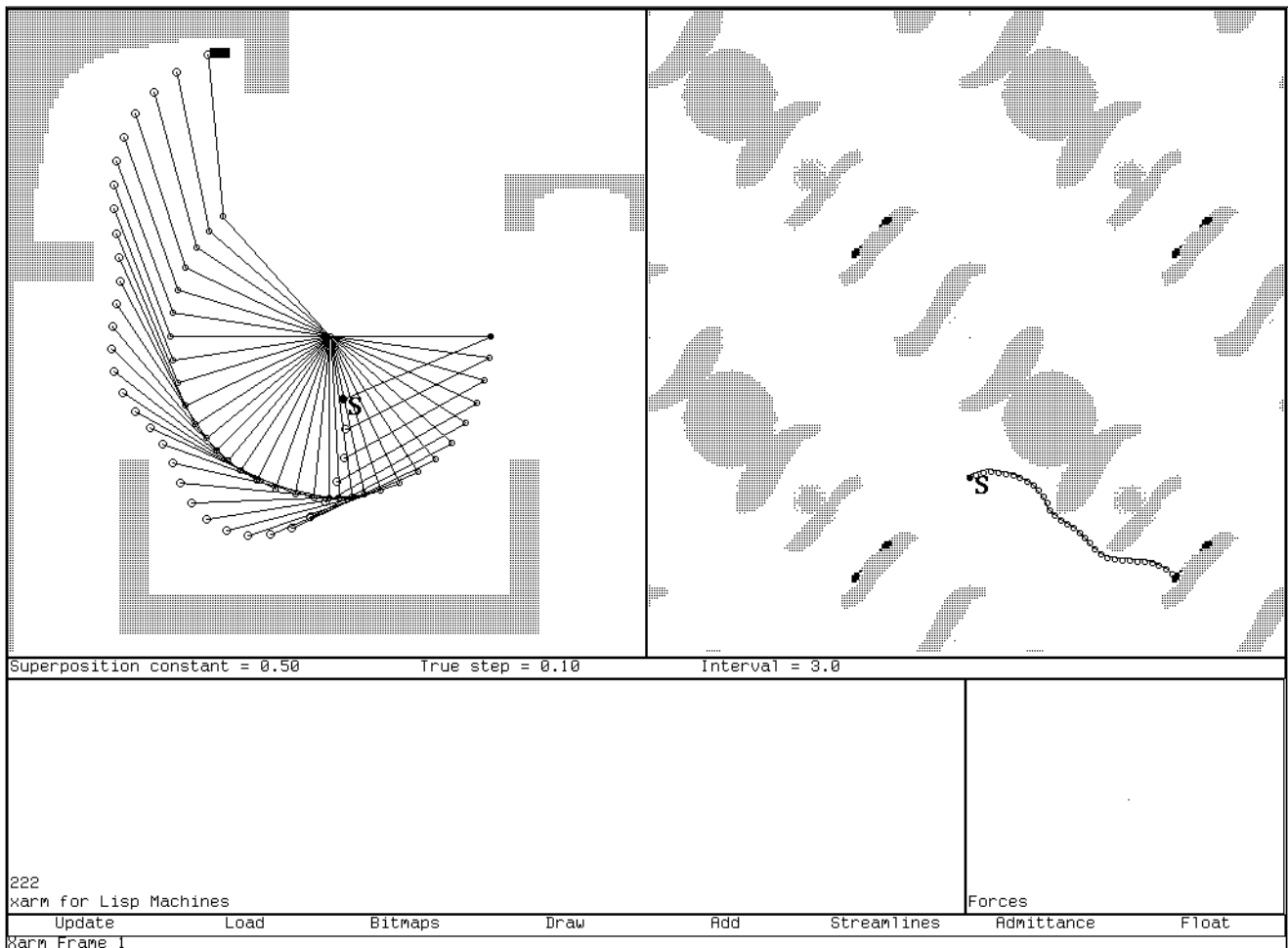
10

Figure 5: Path using $k = 0.99$ (Dirichlet).
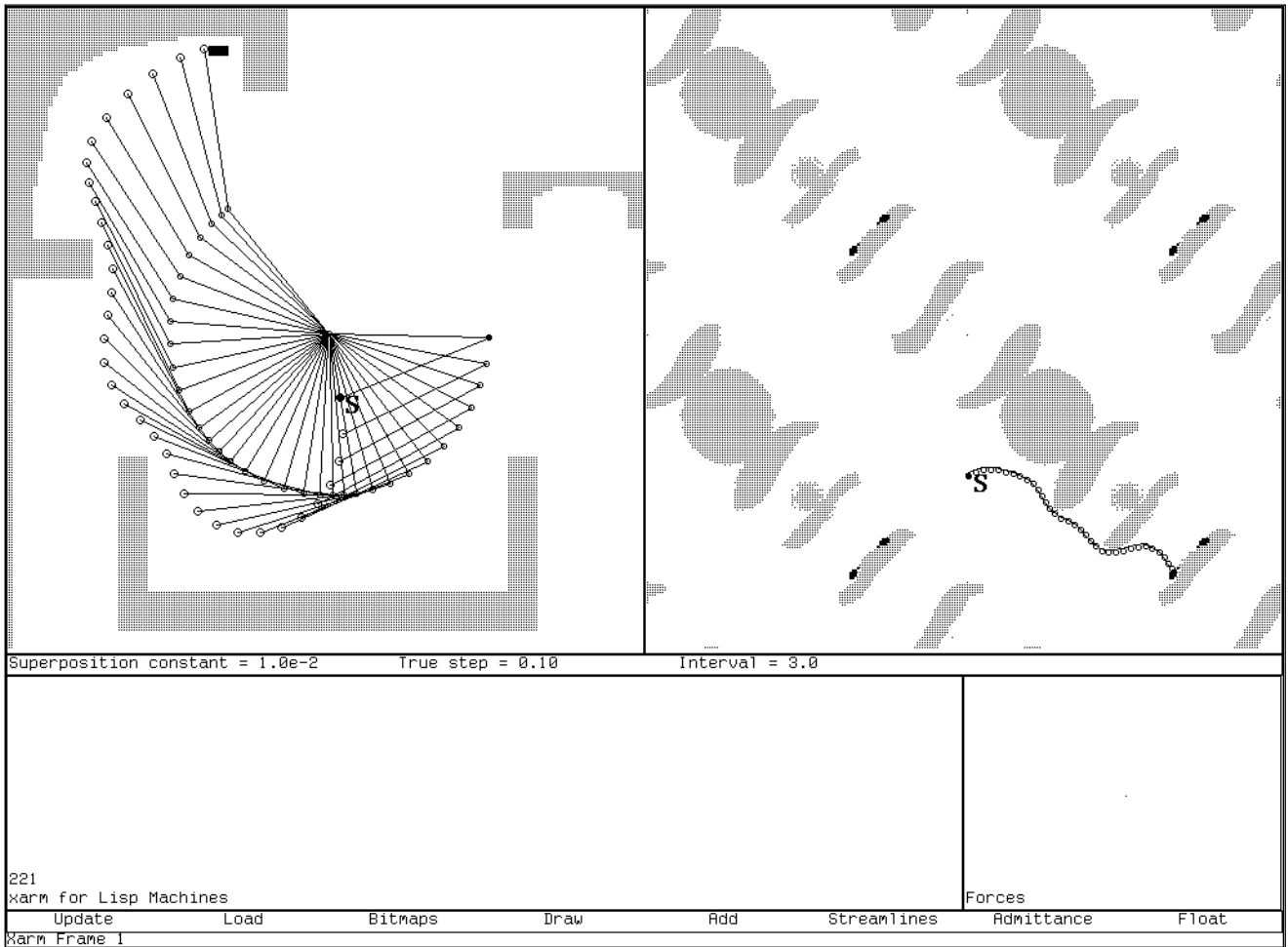
Figure 6: Path using $k = 0.5$ (blend).

12

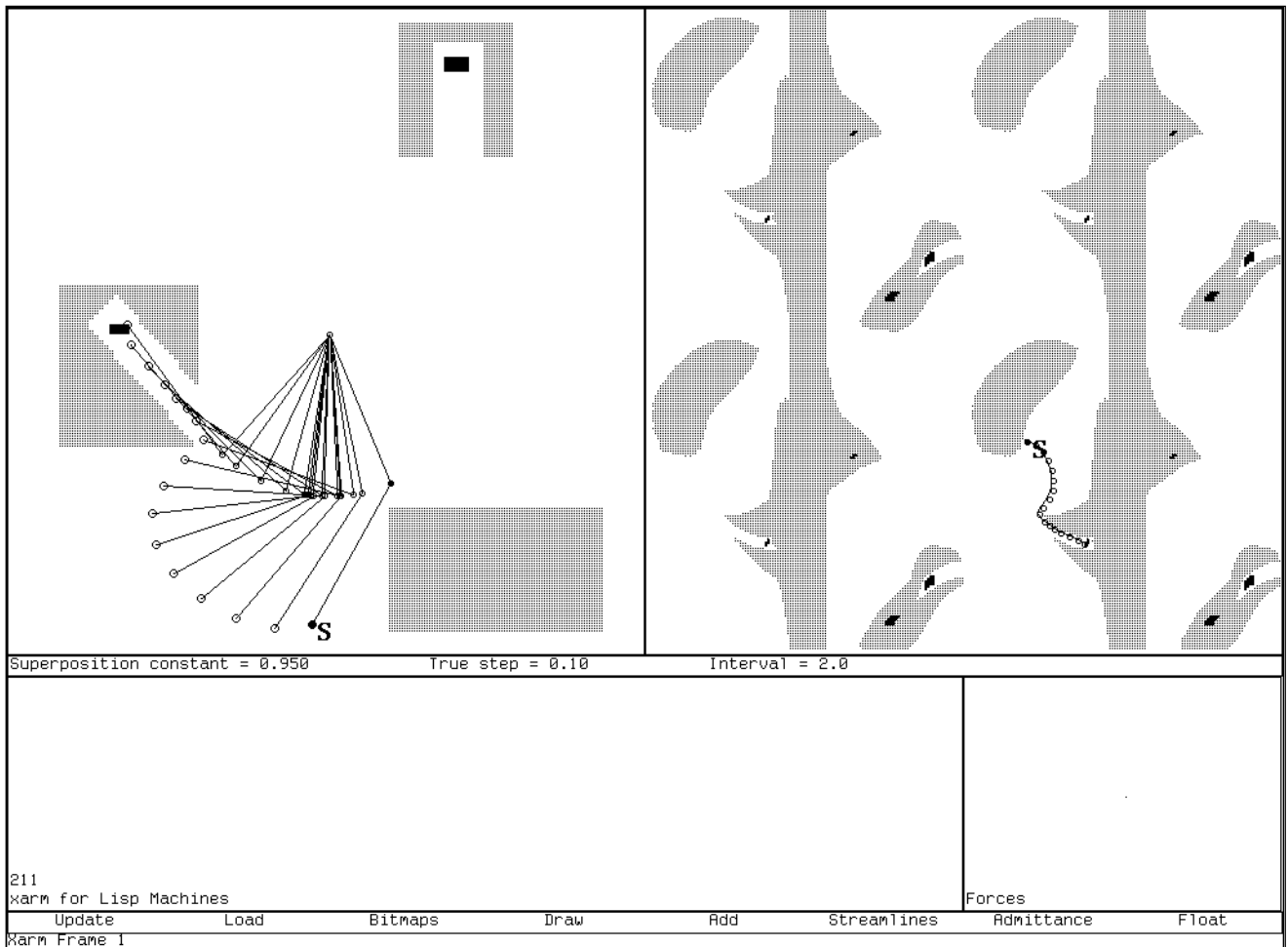Figure 7: Path using $k = 0.01$ (Neumann).

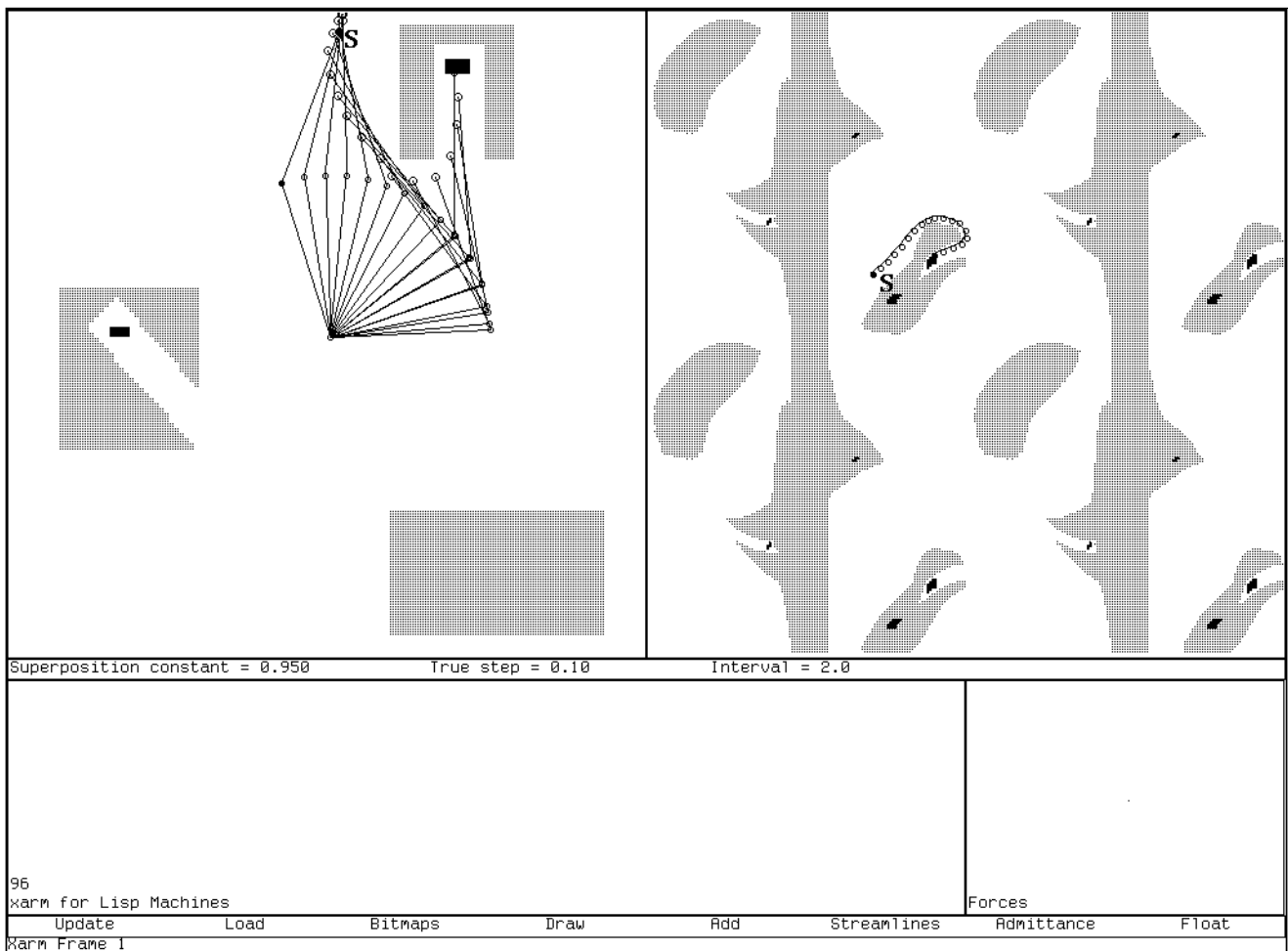Figure 8: Two-goal problem: first path.
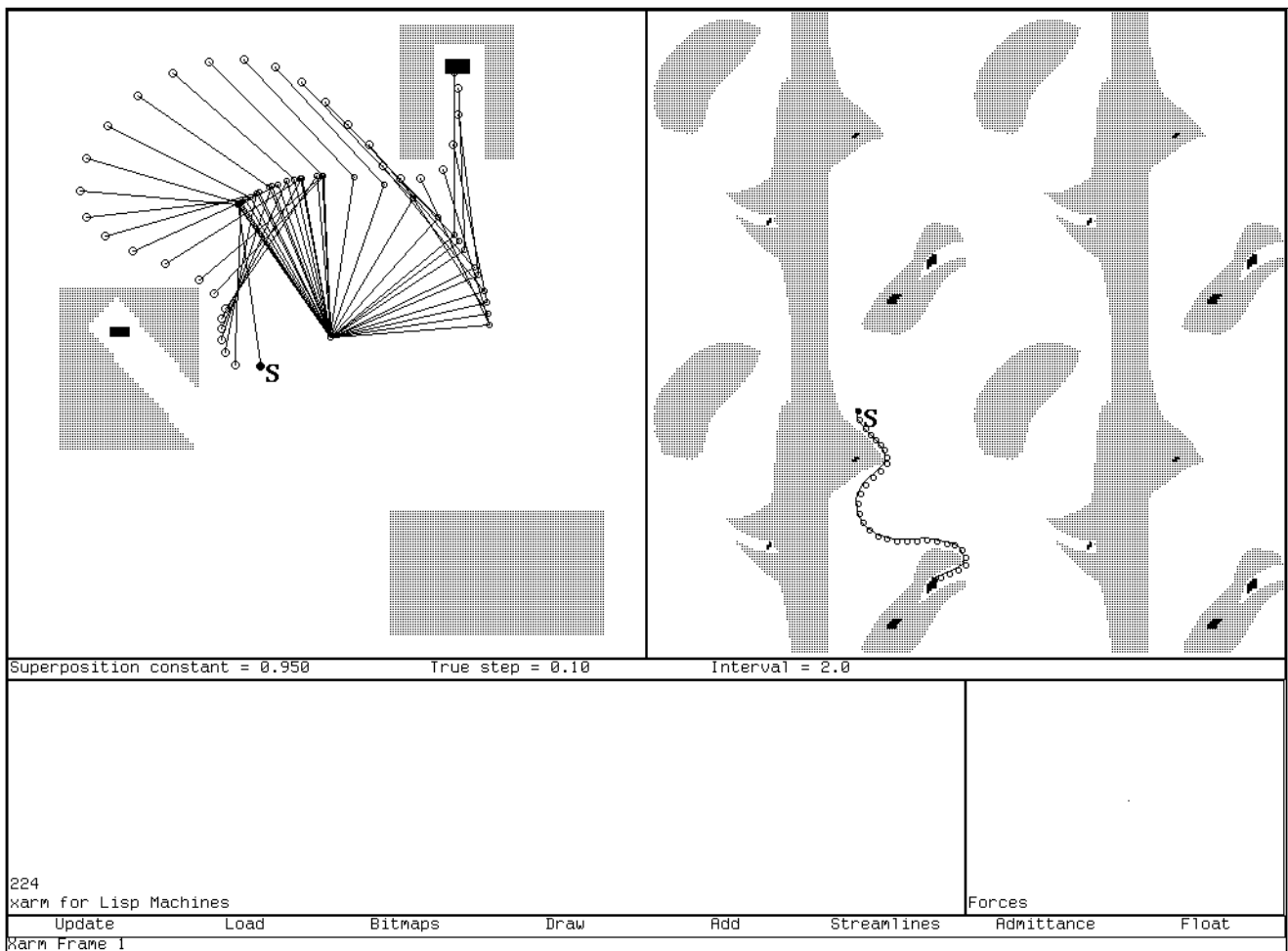
Figure 9: Two-goal problem: second path.

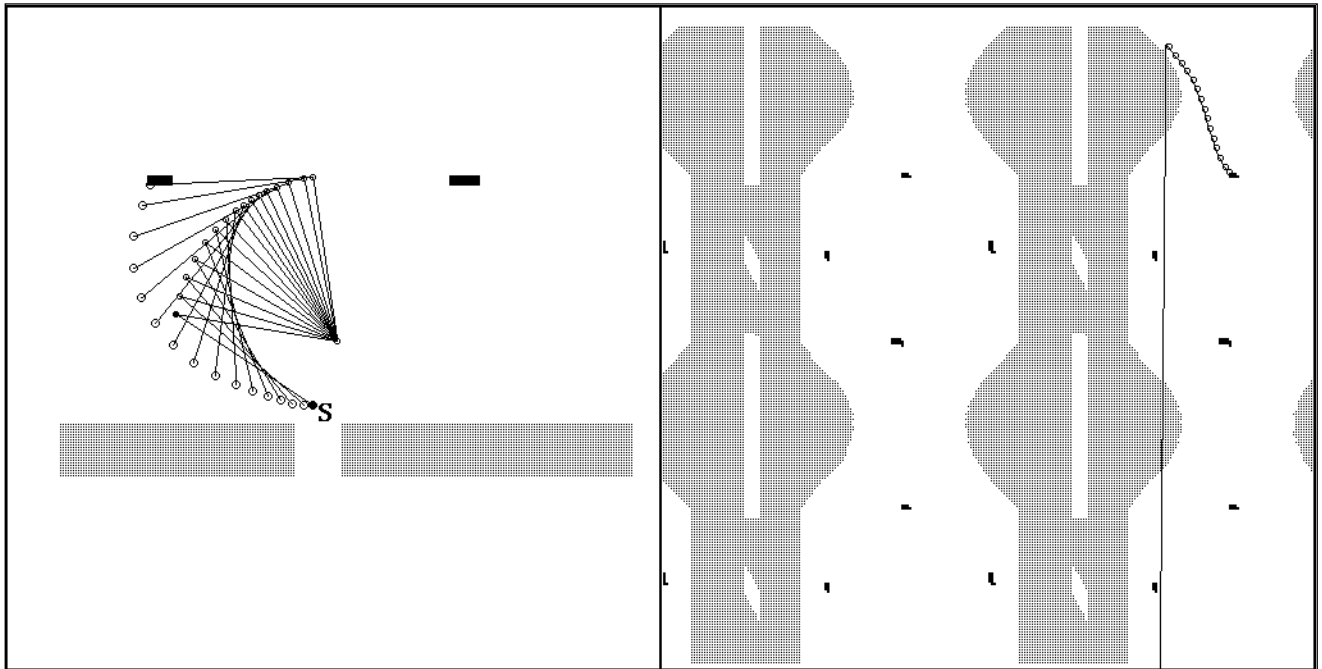Figure 10: Two-goal problem: third path.

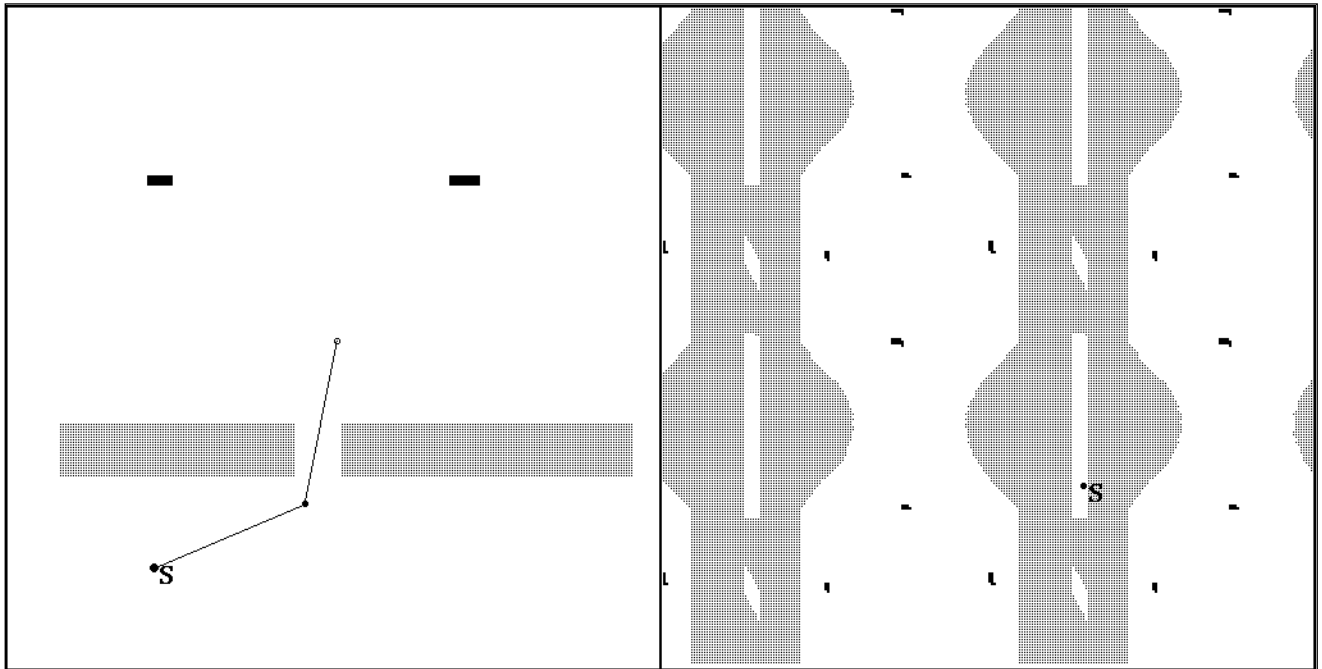Figure 11: Problem B: A two-goal, two-obstacle workspace.

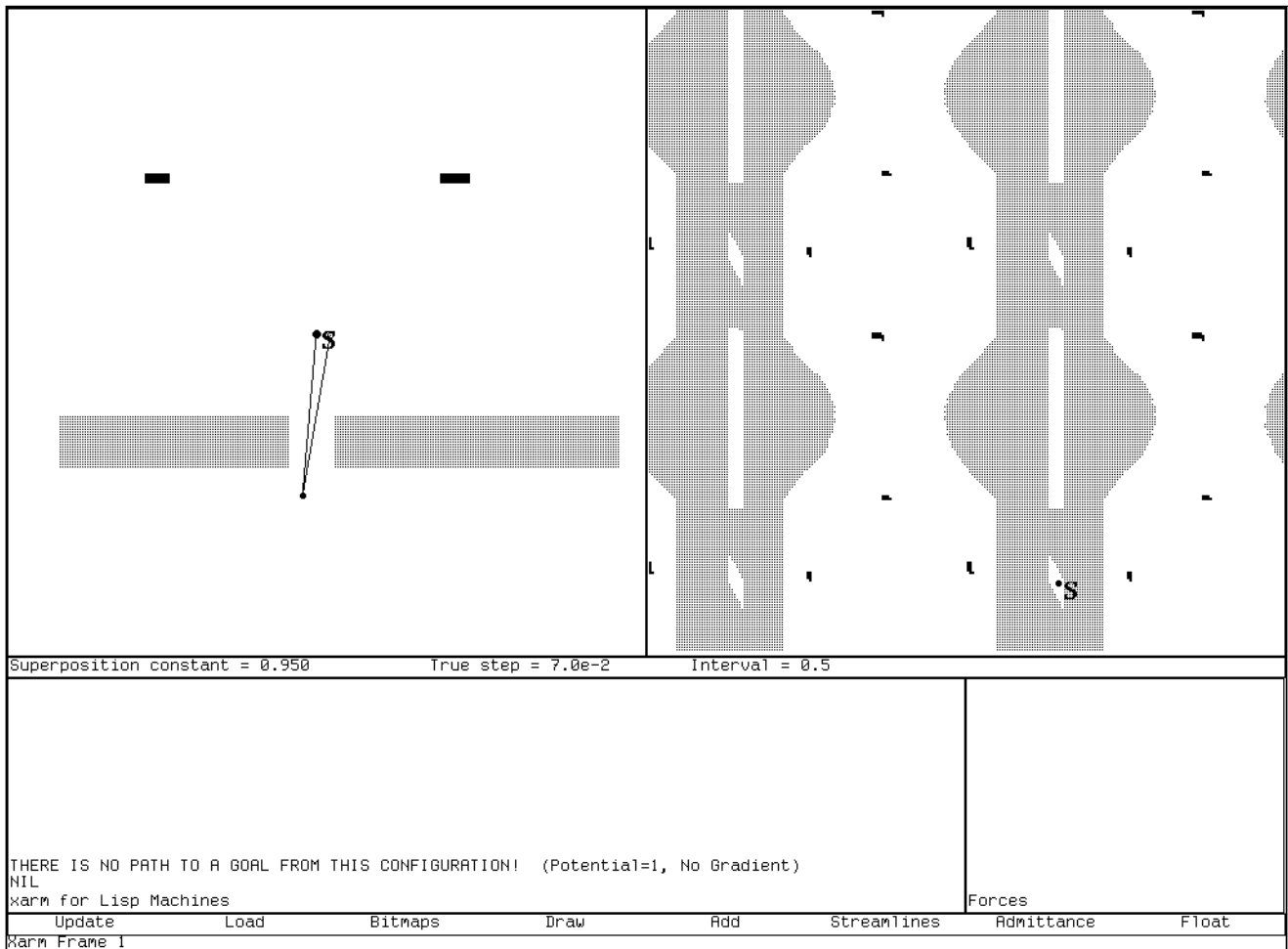Figure 12: A start configuration for problem B which has no path to a goal.

Figure 13: Another start configuration for problem B which has no path to a goal.

5 Harmonic Control

Harmonic functions can be computed using relaxation over a grid (see [1, 13]). Configuration space is digitized and represented in this grid so that obstacles are held at a fixed high potential, while goal points in configuration space are held at a low potential. The relaxation only occurs over points in freespace. Such relaxation methods always converge. After convergence at any point in the freespace portion of the grid, the value of the harmonic function is the average of the values at the grid neighbors. Obviously, this implies that within the freespace region there will be no minima or maxima. These can only occur at goals or obstacles. The above description refers to Dirichlet boundary conditions, where boundary potentials are fixed, e.g.:

$$\phi(\mathbf{x} \mid_{\mathbf{boundary}}) = c$$

Neumann boundary conditions were employed by Tarassenko and Blake [2], and can be computed (digitally) in a similar fashion, although grid values in this case are *reflected* across the boundaries, rather than fixed, e.g.:

$$\mathbf{n} \cdot \nabla \phi(\mathbf{x} \mid_{\mathbf{boundary}}) = 0$$

where $\mathbf{n}$ is the boundary surface normal. The means of computing harmonic functions are described in much more detail in [1], [13], and [2], and their theory is more fully described in [4] and [14], among others.

In this section, harmonic functions are used in conjunction with velocity-based control to implement a robot system which is flexible in coping with errors in the environment model and in robot positions. The method relies on a key property of harmonic functions: Namely, that every streamline of a harmonic function will always reach a desired goal (global minimum) region [9]. This follows directly from the fact that, unlike most other potential field methods, harmonic functions exhibit no spurious local minima [1].

Velocity-based control schemes have received much recent attention. The generalized damper and other passive control models have nice stability properties, and are hence quite attractive [10, 11, 12, 15]. In this paper, we explore velocity based control in conjunction with harmonic functions. The scheme described here attempts to drive the manipulator around any unexpected surfaces (and thence to the goal) in response to external forces generated by contact with those surfaces.

Forces which arise from contact will have a normal component and a friction component. If the contact force lies within the friction cone, sticking will occur. This is characterized by the property that sensed forces will be nearly tangent to the streamline being followed. In sliding contact, the velocity vector will be outside the friction cone, and therefore the force vector will not be colinear with the velocity. In this paper we will discuss a strategy for trajectory modification

that is especially suited for use with harmonic functions. This strategy is called "equipotential-following". It entails modifying the command velocities of a manipulator in the presence of forces so that the manipulator tends to follow the equipotential lines of the controlling harmonic function.

The streamline solution represents a c-space trajectory which avoids all modeled obstacles [1]. The equipotential surface (an $n-1$ dimensional submanifold of c-space) is everywhere orthogonal to the streamlines which cross it [9]. In this sense, the equipotential surface is an extrapolation of the neighboring model geometry. Therefore, we can use the equipotential surface to define an $n-1$ dimensional frame for admittance control. Normally, without any sensed forces, the controller produces velocities which are tangent to a streamline. In a completely modeled environment, these streamlines will always terminate at a goal. However, if a sensed force is detected, the controller switches discretely to follow the equipotential surface. The sensed force projected onto the equipotential surface is a heuristic for searching for alternative streamlines. The controller effectively neutralizes friction and thus reduces the probability of jamming while executing a c-space trajectory in an uncertain environment.

6 Equipotential Following

This technique has much in common with traditional admittance techniques, such as that described by Peshkin [12]. In our context, the harmonic function which is used for generating robot paths is usually referred to as the potential function (see Courant and Hilbert [4]), which is often denoted by ϕ. Streamlines are generated by following the gradient of this potential function, and follow constant values of the *stream function* (usually denoted by ψ). The equipotential surfaces of a harmonic function are always orthogonal to these streamlines.

The presence of contact forces implies that the current streamline is blocked due to an unexpected obstacle. In order to make progress to a goal configuration, then, the robot must move itself off the current streamline and onto a streamline that is free of obstacles. This suggests the use of equipotential surfaces as a means of finding alternative free streamlines.

The method of following equipotential lines results in velocity commands which are tangent to the harmonic function's equipotential surface at the current robot location. Since the equipotential surface is orthogonal to the gradient, the latter actually defines the vector subspace from which we must choose a new velocity command. Let ϕ be the potential function, and ψ be the corresponding stream function, and let $v_\psi = c\nabla\phi$ be a velocity vector along the streamline (where c is a speed-limiting scale factor). We wish to determine a new velocity vector, v_ϕ, which is tangent to the equipotential surface. This velocity is therefore constrained by $\nabla\phi \cdot v_\phi = 0$. The velocity command v_ϕ is still underconstrained; we must choose a particular direction within the equipotential surface. If the contact force has a component which is in the subspace defined by the equipotential surface,

21

then we may use the following to constrain the velocity command:

$$v_\phi = c(v_\psi \times (f \times v_\psi))$$ (1)

where c is again a speed-limiting scale factor. This vector represents the direction of search along the equipotential surface for an alternative streamline, as computed from the original command velocity v_ψ and contact force f.

Figure 14 illustrates the effect of equation 1 on a sliding contact. In this figure, the effector is traveling along a streamline (in joint space), and encounters an obstacle (shaded). Its command velocity would normally be v_ψ in order to continue following the streamline. Force f is observed, consisting of a normal component f_n and a friction component f_f. The new command velocity direction is derived effectively by projecting f onto the equipotential surface, resulting in v_ϕ.

7 Examples

A system has been constructed which simulates velocity control of a 2-link revolute robot arm. The arm is driven by a harmonic function computed over a configuration space grid. This harmonic function is a linear combination of the Neumann and Dirichlet solutions, with a heavy weight (0.98) on the Dirichlet solution. The Neumann component was used to eliminate any precision problems (see [2]). In Figures 15 - 19, the left hand pane shows the robot in its Cartesian workspace. Obstacles are shaded gray, and goals are solid black. In the right hand pane, the robot configuration space is shown, along with the configuration space representation of the obstacles (shaded) and the goals (black). In this system, the robot is treated according to Salisbury's whole-arm manipulator concept. That is, the links themselves are effectively used as force sensors. All forces are ultimately represented as torques in joint space.

The first example shows a single-goal problem. In Figure 15, the robot is started at a configuration and allowed to proceed normally to the goal. In Figure 16, an obstacle has been placed in the rectangular container near the bottom of Cartesian space. Note that this obstacle has not been mapped into configuration space, and that the harmonic function being used was computed in its absence. Thus, many streamlines in the vicinity of this obstacle will travel through it. In this case, the robot is started at the same configuration as seen in Figure 15. Here, however, the forces (actually torques) generated by contact with the new surface prevent the robot from moving further along its original streamline. Once forces (torques) are sensed, a new velocity vector is computed which takes the robot along the equipotential surface at its current configuration, and in a direction which will reduce or eliminate the torques.

Figure 17 shows a new problem with a chair-like obstacle. This figure shows free movement of the robot from its start configuration to the goal. In Figure 18, an unexpected obstacle has
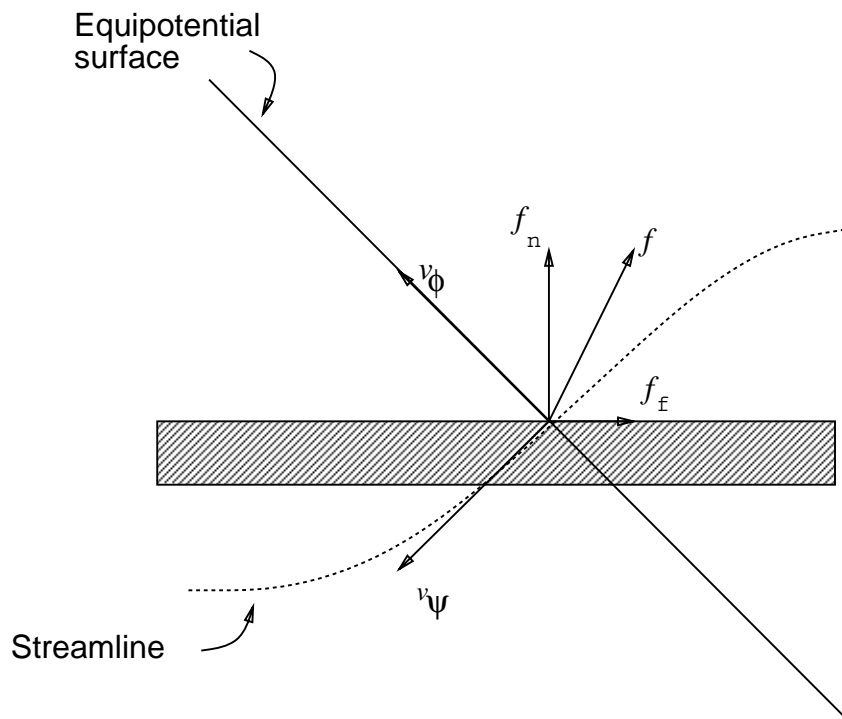
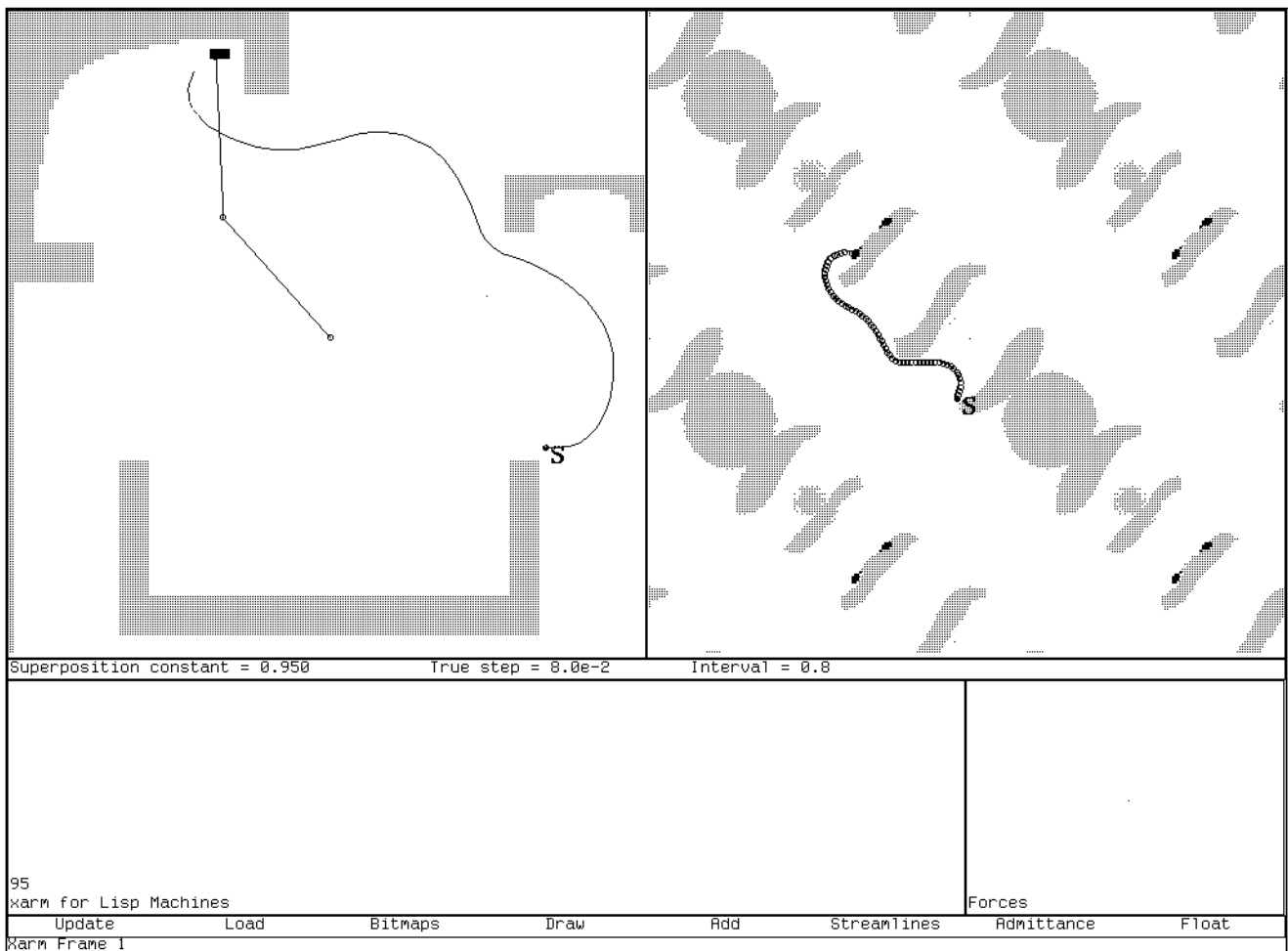Figure 14: Diagram of forces and velocities at a contact.

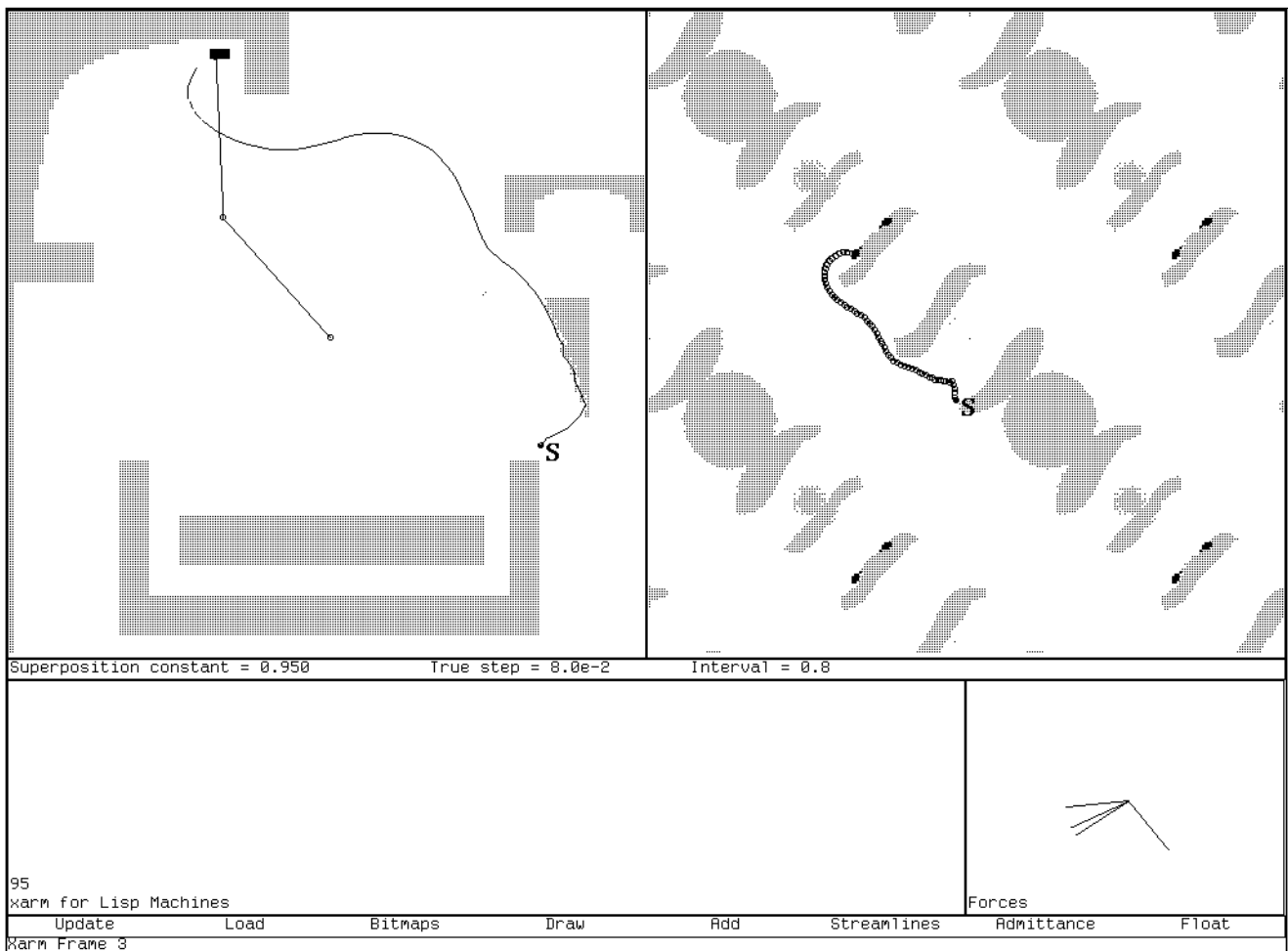Figure 15: Problem 1: A single-goal problem.

Figure 16: Problem 1, but with a new obstacle, and without harmonic function recomputation.

been placed along this path. Note that in this trajectory, the link (rather than the end effector) is deflected. The robot point in configuration space is then deflected along equipotential lines to successively new streamlines, until a free streamline is found which results in free progress to the goal. In the top right corner of the obstacle, it appears as if link 2 is piercing the obstacle. This is a graphic effect due to discretization error and low resolution in the Cartesian space, rather than a problem in the control algorithm.

8   Conclusion

In conclusion, harmonic functions exhibit several properties which are useful for robot control:

- lack of spurious minima

- completeness up to discretization

- behavioral variability through superposition

- robust control in the presence of environmental model error

- computation of surface normals as a by-product

In conjunction with the results of [2], this demonstrates that harmonic functions provide a very attractive mechanism for robot control.

The strategy described in this paper provides a robust means of controlling a manipulator in the presence of external forces. These forces are a manifestation of unexpected obstacles in the manipulator workspace. In general, the scheme described here will still result in the manipulator's progress to the goal configuration in spite of unexpected errors. Moreover, even some gross errors in the robot system's world model can be circumvented using this scheme.

The techniques in this paper can also be used to accumulate information about the environment through force sensing. This strategy cannot overcome the type of singularity generated when the sensed force is precisely aligned with the streamline tangent. Persistent external forces at certain configurations will most likely represent new, previously unsensed obstacles. Hence, they can be used to update the Cartesian and configuration space maps of the environment. For example, Figure 19 shows how the deflections seen in Figure 18 produced new obstacle points in configuration space. This information can subsequently be used to generate a new harmonic function which takes these new obstacles into account. The harware implementation described by Tarassenko and Blake [2] can provide this update very quickly.
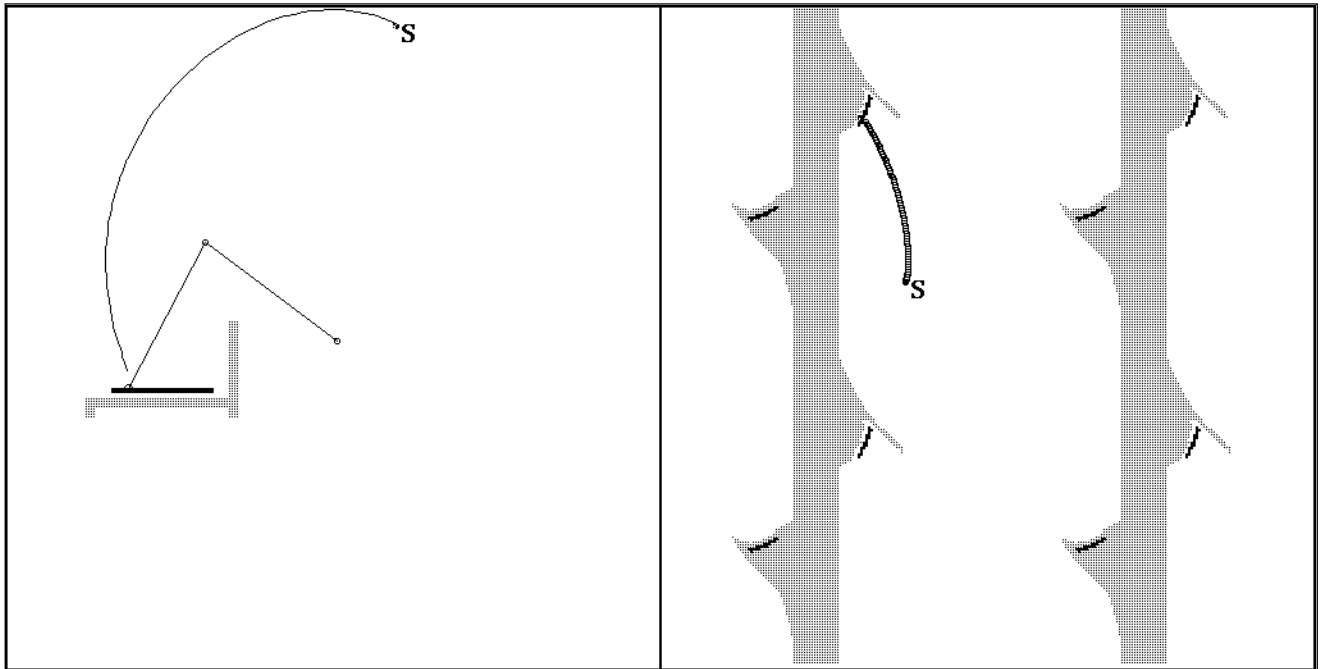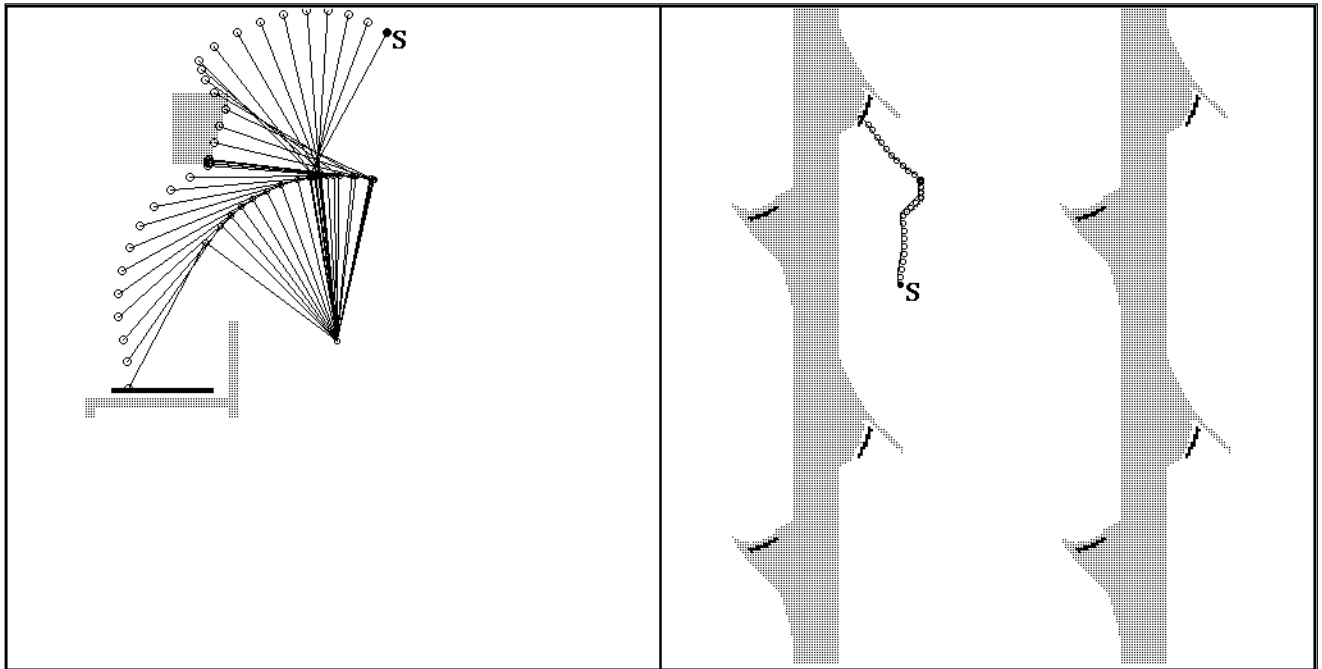
26

Figure 17: Problem 2

Figure 18: Problem 2 with a new obstacle and resulting robot positions superimposed.

Even in the case where fast computation of harmonic functions is not possible, one can still drive the manipulator in parallel with the harmonic function calculation. Since new obstacles essentially place a high potentials at points in the configuration space grid, these potentials will result in wavefronts which will propagate outward from those points until convergence is again achieved.

The authors would like to thank Kamal Souccar, who implemented the 2-link robot simulator.

References

[1] C. I. Connolly, J. B. Burns, and R. Weiss. Path planning using Laplace's Equation. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 2102–2106, May 1990.

[2] L. Tarassenko and A. Blake. Analogue computation of collision-free paths. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 540–545. IEEE, April 1991.

[3] Elon Rimon and Daniel E. Koditschek. Exact robot navigation using cost functions: The case of distinct spherical boundaries in E^n. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pages 1791–1796. IEEE, April 1988.

[4] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 2. John Wiley and Sons, New York, 1989.

[5] Ruel V. Churchill, James W. Brown, and Roger F. Verhey. *Complex Variables and Applications*. McGraw-Hill Book Company, 3rd edition, 1976.

[6] Paul Besl and Ramesh Jain. Intrinsic and extrinsic surface characteristics. In *Proceedings of the 1985 Conference on Computer Vision and Pattern Recognition*, page 226. IEEE, June 1985.

[7] John Ross Stenstrom. Finding points of high curvature and critical curves from a volume model. In *Proceedings of the 1985 Conference on Robotics and Automation*, page 436. IEEE, April 1985.

[8] C. I. Connolly. Cumulative generation of octree models from range data. In *Proceedings of the First International Conference on Robotics and Automation*, page 25. IEEE, March 1984.

[9] Rolf H. Sabersky, Allan J. Acosta, and Edward G. Hauptmann. *Fluid Flow*. MacMillan Publishing Company, 2nd edition, 1971.
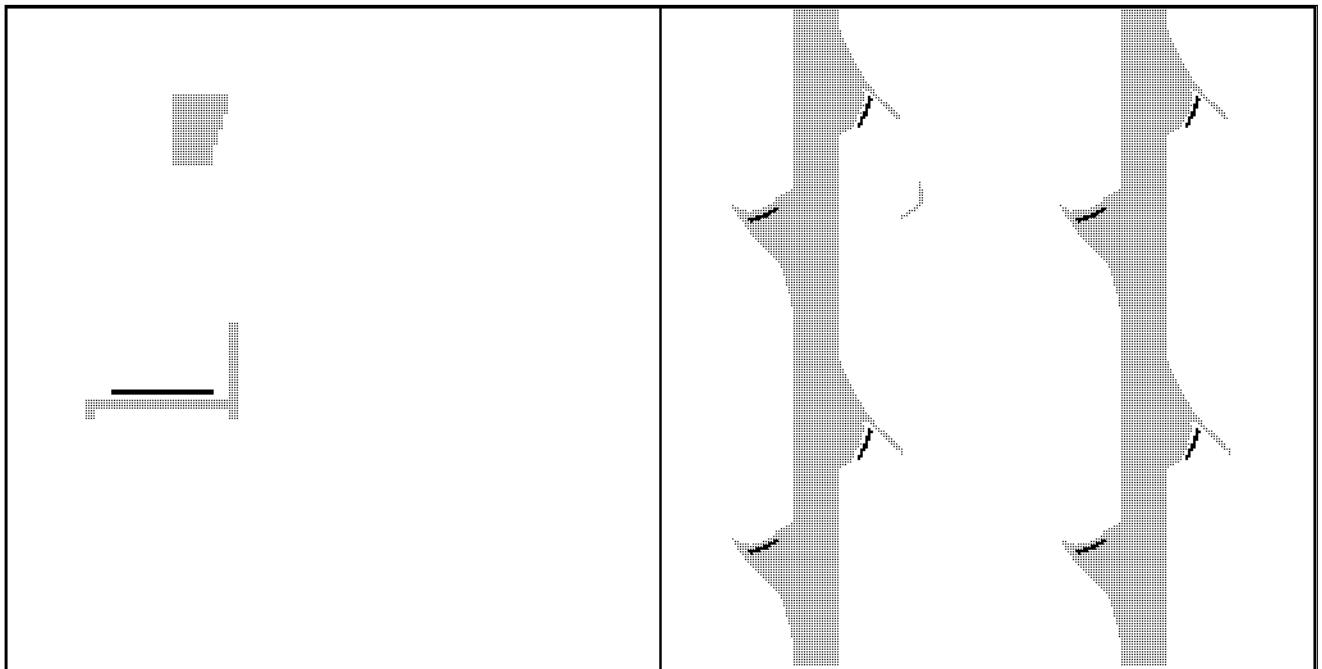
Figure 19: The mapping which resulted from force sensing of the new obstacle in problem 2. The new c-space obstacle points are circled.

[10] Daniel E. Whitney. Force feedback control of manipulator fine motions. *Journal of Dynamic Systems, Measurement, and Control*, pages 91–97, June 1977.

[11] Daniel E. Whitney. Historical perspective and state of the art in robot force control. *International Journal of Robotics Research*, 6(1):3–14, Spring 1987.

[12] Michael A. Peshkin. Programmed compliance for error corrective assembly. *IEEE Transactions on Robotics and Automation*, 6(4):473–482, August 1990.

[13] Richard L. Burden, J. Douglas Faires, and Albert C. Reynolds. *Numerical Analysis*. Prindle, Weber and Schmidt, Boston, 1978.

[14] Ruel V. Churchill and James Ward Brown. *Fourier Series and Boundary Value Problems*. McGraw-Hill Book Company, 3rd edition, 1978.

[15] Ambarish Goswami and Michael A. Peshkin. A task-space formulation of passive force control. In *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, pages 95–100. IEEE, August 1991.