Harmonic Control

Christopher I. Connolly
Roderic A. Grupen
Laboratory for Perceptual Robotics,
Computer Science Department,
University of Massachusetts at Amherst *

May 26, 1992

Abstract

Harmonic functions provide a method for implementing potential-field path planning. Since they produce smooth vector fields which are free of spurious local minima, they also provide an ideal mechanism for implementing velocity-based control. This paper describes the implementation of velocity control using harmonic functions. The resulting control system is termed "Harmonic Control" and is quite robust in the face of geometric error in the environmental model.

1 Introduction

Harmonic functions were introduced in Connolly et al. [1] as a potential-field path planning scheme. In Tarassenko and Blake [2], an analogue circuit was described which provides the means for fast computation of harmonic functions. Several other valuable properties of harmonic functions are described in Connolly [3].

Harmonic functions are solutions to Laplace's Equation in n dimensions:

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

which governs steady-state ideal fluid flow, electrostatic potentials, and steady-state heat distribution. Briefly, harmonic functions exhibit the following valuable properties:

- Completeness up to discretization error
- Lack of spurious local minima
- Linearly superposable
- Robustness with respect to geometrical uncertainty
- Continuity and smoothness of C-space trajectories

Harmonic functions can be computed using relaxation over a grid (see [1, 4]). 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.:

$$\phi(\mathbf{x}\,|_{\mathbf{boundary}}) = c$$

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.:

$$\mathbf{n} \cdot \nabla\phi(\mathbf{x}\,|_{\mathbf{boundary}}) = 0$$

*Portions of this work were funded by the National Science Foundation under grants CDA-8921080 and CDA-8922572.

where $\mathbf{n}$ is the boundary surface normal. The means of computing harmonic functions are described in much more detail in [1], [4], [2] and [3], and their theory is more fully described in [5] and [6], among others.

In this paper, 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 [7]. 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 [8, 9, 10, 11]. 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 [7]. 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.

2 Equipotential Following

This technique has much in common with traditional admittance techniques, such as that described by Peshkin [10]. In our context, the harmonic function which is used for generating robot paths is

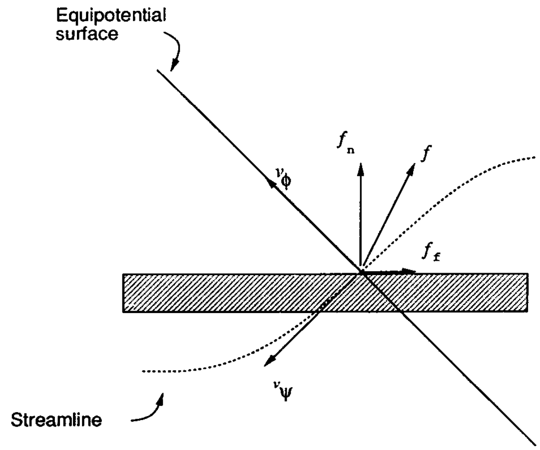

Figure 1: Diagram of forces and velocities at a contact.

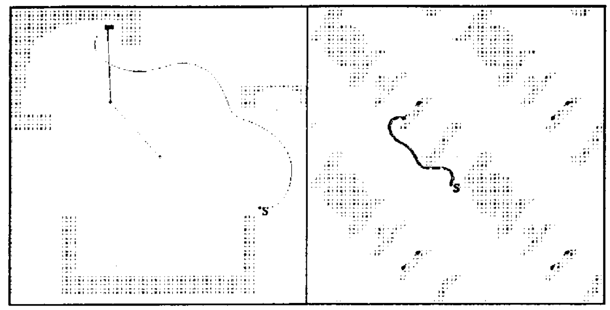

Figure 2: Problem 1: A single-goal problem.

Figure 3: Problem 1, but with a new obstacle, and without harmonic function recomputation.

usually referred to as the potential function (see Courant and Hilbert [5]), 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, then we may use the following to constrain the velocity command:

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

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 1 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.

3 Experiments

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 2 - 6, 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 2, the robot is started at a configuration and allowed to proceed normally to the goal. In Figure 3, 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 2. 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 4 shows a new problem with a chair-like obstacle. This fig-

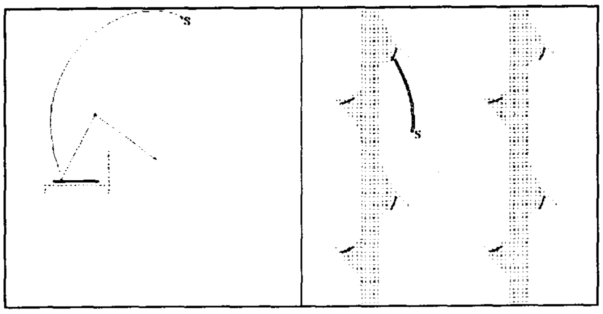

Figure 4: Problem 2

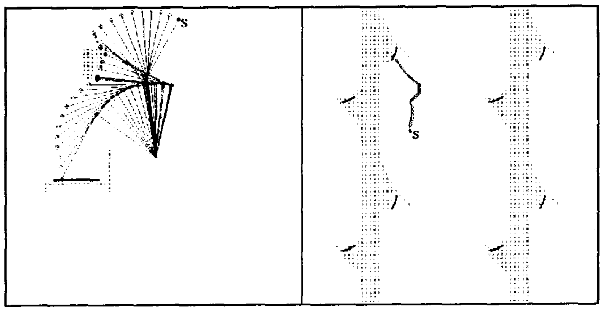

Figure 5: Problem 2 with a new obstacle and resulting robot positions superimposed.

ure shows free movement of the robot from its start configuration to the goal. In Figure 5, an unexpected obstacle has 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.

4 Conclusion

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 6 shows how the deflections seen in Figure 5 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. A harware implementation (such as that described in [2]) can provide this update very quickly.

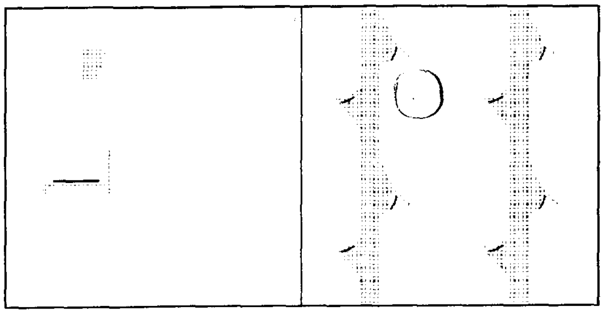

Figure 6: The mapping which resulted from force sensing of the new obstacle in problem 2. The new c-space obstacle points are circled.

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] C. I. Connolly and R. Grupen. Applications of harmonic functions to robotics. Technical Report 92-12, COINS Department, University of Massachusetts, February 1992.

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

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

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

[7] Rolf H. Sabersky, Allan J. Acosta, and Edward G. Hauptmann. *Fluid Flow.* MacMillan Publishing Company, 2nd edition, 1971.

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

[9] 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.

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

[11] 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.