On the 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 [†]

October 9, 1993

Abstract

Harmonic functions are solutions to Laplace's equation. Such functions can be used to advantage for potential-field path planning, since they do not exhibit spurious local minima. Harmonic functions are shown here to have a number of properties which are essential to robotics applications. Paths derived from harmonic functions are generally smooth. Harmonic functions also offer a complete path planning algorithm. We show how a harmonic function can be used as the basis for a reactive admittance control. Such schemes allow incremental updating of the environment model. Methods for computing harmonic functions respond well to sensed changes in the environment, and can be used for control while the environment model is being updated.

[*] *Journal of Robotic Systems*, 10(7):931-946

[†] This work is supported in part by the National Science Foundation under grants CDA-8922572, IRI-9116297 and IRI-9208920.

Contents

1 Introduction

Potential fields were promoted by Khatib [4] for robot path planning. Other authors [5, 6, 7, 8, 9, 10] have used a variety of potential functions for similar purposes. Unfortunately, the usual formulations of potential fields for path planning do not preclude the spontaneous creation of minima other than the goal. The robot can fall into these minima and achieve a stable configuration short of the goal [4, 8, 11, 12, 13] . Koditschek [10] showed that this need not be the case in certain types of domains.

Connolly, et al. [2] , and independently Akishita, et al. [14] described the application of harmonic functions to the path-planning problem. Harmonic functions are solutions to Laplace's equation. This paper describes harmonic functions and their application to various robot control problems.

Harmonic functions are shown here to have several useful properties which make them well suited for robotics applications:

- Completeness (up to discretization error in the environment model)
- Robustness in the presence of unanticipated obstacles and errors
- Ability to exhibit different useful modes of behavior
- Rapid computation (computed as voltages in a resistive grid [15])

In addition, harmonic functions can provide fast surface normal computation and geometric extrapolation.

The method described here is a robust form of reactive path planning. Other techniques [13, 16] require substantial off-line computation which prohibits the system from reacting well to unexpected changes in the environment. In contrast, harmonic functions allow the model of the environment to be updated incrementally. Therefore, incomplete or non-stationary environmental models can be modified on-line during execution.

2 Configuration Space

The techniques described here require that the effector be mapped to a point in configuration space [17] . The effector acts in Cartesian space, and its configuration can be expressed in terms of joint positions. The path planning problem is then posed as the construction of an obstacle-avoiding path from a start point to a goal point in configuration space (C-space).

A bitmap representation of the workspace suffices for computing the desired harmonic function. Figure 1 illustrates the conversion process: a Cartesian space grid is constructed which contains information about obstacles and goals. Two bits are used for each grid point: one bit designates obstacle points, while the other represents goal regions. Zero is used to denote freespace. Both the obstacle and the goal regions can be arbitrarily shaped up to discretization. C-space bitmaps are first initialized to 0 (freespace). Then for each C-space bitmap pixel, the equivalent pixels in the Cartesian bitmap are checked using the manipulator forward kinematics. If any of the corresponding Cartesian pixels are occupied, then the C-space bitmap pixel is marked as occupied.

Figure 2 shows the result of this mapping for a simple robot workcell. A rectangular object is illustrated inside the workspace of a 2-dof planar manipulator. The C-space for this problem wraps around at the boundaries. There are multiple C-space goals, and there are inaccessible goal configurations embedded within obstacles.

3 Harmonic functions

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

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

In the case of robot path planning, the boundary of Ω (denoted by $\partial\Omega$) consists of the boundaries of all obstacles and goals in a C-space representation. Harmonic functions satisfy the min-max principle: spontaneous creation of local mimima within the solution region is impossible. The Gauss Integral Theorem applied to harmonic functions (Courant and Hilbert [18]) states that for a harmonic function ϕ, the following integral holds:

$$\int \cdots \int_{\partial\Gamma} \frac{\partial\phi}{\partial\underline{n}} dS = 0 \tag{2}$$

which is an integral of normal derivatives over the closed boundary ($\partial\Gamma$) of an arbitrary region $\Gamma \subset \Omega$. This theorem states that there is a balance between inward and outward flow on the boundary of any volume within the solution region (not including obstacles or goals). Thus, there is always a "way out" from any point or region. This theorem holds for any number of dimensions.

The gradient vector field of a harmonic function has zero curl, and the function itself obeys the min-max principle. Hence, the only types of critical points which can occur are saddle points. The exit from such a critical point can be found by performing a search in the neighborhood of that point. Moreover, any perturbation of a path (also referred to as a *streamline*) from such points results in a path which is smooth everywhere.

3.1 Numerical solutions of Laplace's equation

Numerical solutions for Laplace's equation are readily obtained from Finite Difference methods [19] . Let $\phi(x, y)$ be a function which satisfies Laplace's equation, and let $u(x_i, y_j)$ represent a discrete regular sampling of ϕ on a grid. A Taylor series approximation to the second derivatives can then be used to derive the discrete form of Laplace's equation:

$$h^2\phi(x_i, y_j) = u(x_{i+1}, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1}) - 4u(x_i, y_j) \tag{3}$$

where h is the grid spacing, and may be set to 1 in this application. Equation 3 on the grid defines a linear system of equations. The variables are the values for Laplace's equation in the free portions of the grid. Relaxation methods are used to solve such systems digitally.

Equation 3 can be extended to higher dimensional grids in the obvious way. Techniques such as Jacobi iteration, Gauss-Seidel iteration or Successive Over-Relaxation (SOR) can be used [19] . For Laplace's equation, these methods essentially consist of repeatedly replacing each grid element's value with a weighted average of its neighbors, using a $2n$-neighborhood (manhattan neighbors) for n dimensional regions. Although these methods compute function values on a grid, multilinear functions are used to interpolate between grid points, since such functions are harmonic and smooth.

3.1.1 Relaxation Methods

In its simplest form, Jacobi iteration for Laplace's Equation replaces every non-boundary $u(x_i, y_j)$ with the average of its neighbors' values *simultaneously.* This scheme can be written as:

$$\begin{aligned} u^{(k+1)}(x_i, y_j) = \frac{1}{4}(\quad & u^{(k)}(x_{i+1}, y_j) + u^{(k)}(x_{i-1}, y_j) + \\ & u^{(k)}(x_i, y_{j+1}) + u^{(k)}(x_i, y_{j-1})) \end{aligned} \tag{4}$$

where k is the iteration number. On a SIMD[1] machine, an iteration over the grid can be accomplished in one parallel step using only one solution variable per processing element. Jacobi iteration generally requires a higher number of iterations for convergence than Gauss-Seidel or SOR. On the other hand, it is quite effective on SIMD architectures.

Gauss-Seidel iteration is similar to Jacobi iteration, except that the iteration numbers for neighboring values are mixed. This is a natural scheme to use in a single-processor system:

$$\begin{aligned} u^{(k+1)}(x_i, y_j) = \frac{1}{4}(\quad & u^{(k)}(x_{i+1}, y_j) + u^{(k+1)}(x_{i-1}, y_j) + \\ & u^{(k)}(x_i, y_{j+1}) + u^{(k+1)}(x_i, y_{j-1})) \end{aligned} \tag{5}$$

One can use a single solution array, replacing each element by the average of its neighbors. Half of these neighbors are from iteration $k+1$, while the other half are from iteration k.

The method of Successive Over-Relaxation (SOR) converges more rapidly than Gauss-Seidel or Jacobi iteration. It is therefore used in most of the examples shown in this paper. The recurrence relation for SOR on Laplace's equation is:

$$\begin{aligned} u^{(k+1)}(x_i, y_j) = u^{(k)}(x_i, y_j) + \frac{\omega}{4}(& u^{(k+1)}(x_{i+1}, y_j) + u^{(k+1)}(x_{i-1}, y_j) + \\ & u^{(k)}(x_i, y_{j+1}) + u^{(k)}(x_i, y_{j-1}) - 4u^{(k)}(x_i, y_j)) \end{aligned} \tag{6}$$

[1] Single Instruction, Multiple Data.

where k represents the iteration number, and ω is the acceleration constant. The value of ω is chosen to speed the rate of convergence; it is 1.8 for the examples shown here. See [19] for a more detailed discussion of acceleration constants.

Finally, the Full Multigrid Method appears to be the fastest digital technique for computing harmonic functions. This method has an asymptotic running time of $O(N)$, where N is the number of nodes in the grid [20] . This method compares favorably with other complete path-planning techniques (e.g., distance propagation).

3.1.2 Boundary conditions – Dirichlet and Neumann

In this discussion, we will use the following restricted forms of the Dirichlet and Neumann boundary conditions. The following is a Dirichlet boundary condition:

$$\phi|_{\partial\Omega} = c \tag{7}$$

where c is a constant. In this case, the boundary is held fixed at a particular potential. For our purposes, c can be set to 1. Solutions derived from this boundary condition will be denoted by ϕ_D. Because the boundary condition is a fixed, high potential, the flow must be along the outward normal of the obstacle surface. The gradient of the harmonic solution, therefore, tends to depart from C-space obstacle boundaries.

The restricted form of the Neumann boundary condition can be described as:

$$\left.\frac{\partial\phi}{\partial\underline{n}}\right|_{\partial\Omega} = 0 \tag{8}$$

where $\underline{n}$ is the vector normal to the boundary. The Neumann boundary condition constrains the gradient of ϕ. The condition described in Equation 8 specifies that the flow is *tangential* to the obstacle (i.e., there is no normal component of flow). Solutions derived from this form of Neumann boundary condition will be denoted by ϕ_N. The tangent-flow behavior exhibited by a Neumann solution can result in a tendency for a robot to stay close to C-space obstacle surfaces. Since the Neumann condition still requires a source and a sink for the flow, the outer boundary of the C-space map has been set up as a source.

One can superpose the Dirichlet and Neumann solutions to obtain a harmonic function that exhibits a behavior somewhere between the two. 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:

$$\phi = k\phi_D + (1-k)\phi_N \tag{9}$$

where $k \in [0,1]$. This guarantees that gradients at obstacle boundaries will have a component which is outward normal to the surface, thus avoiding collisions with known obstacles. Moreover, $k < 1$ avoids shallow gradients sometimes found in the Dirichlet solution [2, 3] . The resulting ϕ is harmonic, has no local minima and guarantees collision-free paths.

Figure 3 illustrates a workspace geometry and the paths corresponding to a superposition constant (k) of 0, 0.5, and 1. The resulting manipulator endpoint trajectories are also illustrated in the bottom row of figure 3.

3.2 Completeness

Under certain assumptions regarding the boundary conditions, 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 R^n.
- ϕ is a solution to Laplace's equation on Ω with Dirichlet boundary conditions.
- Streamlines for ϕ are computed using the gradient $\nabla\phi$.
- All boundaries $\partial\Omega$ are held at the obstacle potential $\phi = 1$.
- All goal regions are held at the goal potential $\phi = 0$.

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

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

$\Rightarrow$ If there is no path from $\underline{x}$ to $\underline{y}$, it must be the case that $\underline{x} \in S$ and $\underline{y} \notin S$. 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 $\underline{x}$, and since the neighborhood is arbitrary, any point in S, must be the same, i.e. $\phi(\underline{x}) = 1$. Therefore, ϕ is constant and $\nabla\phi$ at any point in S must be zero.

$\Leftarrow$ If $\nabla\phi(\underline{x}) = 0$ in S, then the derivatives of ϕ must be zero and ϕ must be a constant function. Consider some point $\underline{z} \in S - \partial S$, and assume that $\underline{z} = \underline{y}$ (the goal point). The min-max property requires that the minimum and maximum values of ϕ in any neighborhood of $\underline{z}$ must be attained on the boundary of that neighborhood. Let a given neighborhood of $\underline{z}$ contain points in ∂S. By the min-max property, since ϕ is constant, it must also be true that $\phi(\underline{z}) = 1$ (the value at ∂S). But if $\underline{z} = \underline{y}$, then $\phi(\underline{z}) = 0$, which cannot be the case. Hence $\underline{y}$ 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 in any connected component without goals, the function will converge to a constant (see Courant and Hilbert [18] for a detailed treatment of this property).

3.3 Hardware Implementation

Resistive networks were employed in the 1940s to compute harmonic functions [15] . Tarassenko and Blake (1990)[3] describe a VLSI version of this hardware scheme for path planning. In Tarassenko and Blake (1990) [3] , a programmable resistive grid is described which consists of an array of transistors whose drain-source resistances are governed by gate voltages. The transistor network can be used to set both Dirichlet and Neumann obstacle types. After settling, the voltages at "free" junctions will represent a discrete sampling of a harmonic function (i.e. node voltages will be the average of neighboring node voltages). Given that the computation is performed by voltages settling in a resistive network, this appears to provide the fastest existing method for robot path planning.

The technique, however, has limitations. As the grid size is increased, the settling time will also increase, and voltage differences within the grid will decrease. Thus, there are physical limits to the size (and hence resolution) of a useful resistive grid. Nonetheless, such limits appear to be in the range of hundreds or perhaps even thousands of resistive elements on a side, using currently available VLSI technology. We consider most applications to be well within these limits.

4 Reactive Trajectory Control

Passive control models (such as the generalized damper) which use complete geometric information are robust with respect to bounded uncertainty. These techniques[21, 22, 23, 24] provide a basis for robust planning. Harmonic functions can be exploited to implement an admittance controller as the basis for a reactive path planner. We show that the admittance controller in conjunction with incremental C-space observations and iterative revision of the harmonic function solution is robust with respect to both uncertain and incomplete world knowledge. We assume here that contacts are observable as forces along C-space coordinate directions. Similar schemes have been proposed by Salisbury, et al. for the WAM (Whole Arm Manipulator) [25] , and by Huber and Grupen [26] for contact localization in multifingered hands. Cheung and Lumelsky [27] use a sensor skin on a robot arm to gather similar contact information.

4.1 Trajectory Generation

Let ϕ be a scalar potential function satisfying Laplace's equation, $\nabla^2\phi = 0$, so that $\nabla\phi$ defines the streamlines of ϕ. A simple streamline-following controller can be obtained by using the gradient of the harmonic function as a velocity or displacement for the effector:

$$\underline{\dot{q}} = K_\nabla \nabla\phi$$

where K_∇ is the velocity gain. This form of control has been implemented for a P-50 robot arm. Figure 4 shows an example of the arm using a harmonic function to avoid a box placed in the center of its workspace. Five frames were superimposed in the figure to illustrate the robot's movement.

The goal is a position forward and to the left of the box. In figure 4, the starting configuration placed the end effector to the right and slightly behind the box. The harmonic function for the three-dimensional configuration space was computed from scratch in approximately 40 seconds using a Motorola 68030 processor. Once the harmonic function is computed, paths to the goal can be executed from any point in the workspace. The first three joints were used. The values for the joint angles over time are shown in Figure 5. This path was executed in approximately 5 seconds.

4.2 Trajectory Control

The equipotential surface (an $n-1$ dimensional submanifold of C-space) is everywhere orthogonal to the streamlines of a harmonic function. Streamlines that intersect the equipotential surface represent equivalent alternative trajectories to the goal configuration on the basis of the metric ϕ. Figure 6 illustrates the equipotential surface and the streamline that defines it at the point of contact with an unmodeled C-space obstacle. The obstacle produces a reaction force, $\underline{w}$, on the manipulator.

An admittance controller derived from equipotential surfaces can be developed for use with harmonic function path planning:

$$\underline{\dot{q}}_c = K_\nabla \nabla \phi + K_{EQ} A(\underline{q}) \underline{w} \tag{10}$$

where:

$\underline{q}$ – represents the current state in configuration space (i.e., $(\theta_0, \theta_1, ..., \theta_n)$ for a revolute robot manipulator),
$\underline{w}$ – represents the contact force observation (i.e., $(\tau_0, \tau_1, ..., \tau_n)$ for a revolute arm),
K_∇ = $1/(\epsilon_\nabla + ||\nabla\phi||)$
= normalized gradient descent multiplier for streamline navigation,
K_{EQ} = $1/(\epsilon_{EQ} + | < \nabla\phi, \underline{w} > |)$
= normalized gradient descent multiplier for equipotential surface navigation,
$A(\underline{q})$ – the admittance relation mapping contact forces, $\underline{w}$, to control velocities, $\underline{\dot{q}}_{EQ}$.

The first term of Equation 10 is a normalized gradient descent on the potential ϕ. The second term is an admittance relation designed to react to forces derived from contacts with unmodeled C-space obstacles. The harmonic function is a global interpolant, and as such provides the basis for a hybrid (position/force) controller in which position can be regulated along the streamline and force can be regulated in the orthogonal equipotential manifold.

The design of an admittance matrix involves selecting a C-space velocity $\dot{\underline{q}}_{EQ}$ that navigates across the equipotential manifold in response to contact forces to locate alternative streamlines to the goal. Equation 11 can be used to define a heuristic admittance matrix:

$$\dot{\underline{q}}_{EQ} = -\left[\nabla\phi \times (\nabla\phi \times \underline{w})\right] \tag{11}$$

The corresponding admittance matrix form for Equation 11 can be derived by using the vector product in matrix operator form. For example, in a three dimensional C-space:

$$\begin{aligned}
\dot{\underline{q}}_{EQ} &= A(\underline{q})\underline{w} && (12)\\
&= -(\nabla\phi \times (\nabla\phi \times \underline{w})) && (13)\\
&= -\nabla\phi \times \begin{bmatrix} 0 & -\phi_{q_3} & \phi_{q_2} \\ \phi_{q_3} & 0 & -\phi_{q_1} \\ -\phi_{q_2} & \phi_{q_1} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} && (14)\\
&= \begin{bmatrix} (\phi^2_{q_2} + \phi^2_{q_3}) & -\phi_{q_1}\phi_{q_2} & -\phi_{q_1}\phi_{q_3} \\ -\phi_{q_1}\phi_{q_2} & (\phi^2_{q_1} + \phi^2_{q_3}) & -\phi_{q_2}\phi_{q_3} \\ -\phi_{q_1}\phi_{q_3} & -\phi_{q_2}\phi_{q_3} & (\phi^2_{q_1} + \phi^2_{q_2}) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} && (15)
\end{aligned}$$

where $\phi_{q_1}, \phi_{q_2}, \phi_{q_3}$ are the first, second, and third components of the gradient, respectively.

In this form, the admittance matrix $A(\underline{q})$ is a function of configuration, and represents a reaction to force which will tend to drive the system along equipotential surfaces at any point in C-space. Thus, an appropriate admittance matrix is an attribute of the harmonic function solution.

A system has been implemented which uses the techniques described here for reactive path planning. The system was implemented on a DECstation 5000 using two processes, one for graphics and control, the other for harmonic function computation. This allowed the asynchronous injection of new obstacle information while the harmonic function was being computed.

Figure 7 (left side) shows the system's harmonic function solution for the path of a point effector from its start configuration to the goal. Black squares denote obstacles and the open square is a goal. The effector's path is shown as a series of circles. In the right-hand side of Figure 7, new obstacles are introduced (the open squares below the original obstacle), but the harmonic function is not recomputed. The admittance controller described in this section is successfully used here to "feel" around the new obstacles. The square which produced the sensed forces is shaded.

4.3 Reactive Path Planning

Information about the environment can be accumulated incrementally during path execution. Equipotential surface trajectories are triggered by new, previously unsensed obstacles. Hence, they can be used to update the Cartesian and C-space maps of the environment.

The Jacobi method described above (Section 3.1.1) is well suited for reactive re-planning. Figure 9 illustrates the impulse response for Jacobi iteration. Note that at each iteration, the function is smooth and symmetric (unlike the intermediate results for Gauss-Seidel or SOR). The local shape of the function makes it possible to drive the effector smoothly even when the function has not yet converged. Even though the iterates are not themselves harmonic, they are still smooth and will result in obstacle-avoiding trajectories to the goal.

Figure 8 shows the effect of combining the equipotential admittance controller with continuous harmonic function computation. In this example, the lower channel is completely blocked with unsensed obstacles (open squares), which will eventually require the effector to travel above the obstacle. Upon encountering the previously unsensed obstacles, the system injects these points into the harmonic function environment map. Admittance control is used to escape to other streamlines, while the harmonic function updating tends to drive the system toward the goal. New obstacle boundaries deform the harmonic navigation function continuously toward a solution which circumvents these boundaries. The total running time for this example was 5 seconds.

5 Summary

Harmonic functions provides a robust means of controlling a manipulator to goal states in the presence of obstacles, external forces and environmental uncertainty. Unexpected external forces are a manifestation of poorly modeled obstacles in the manipulator workspace. In general, the scheme described here will insure 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. External forces can also be used to incrementally update the robot system's map of the environment. The use of harmonic functions as described here allows control to continue *concurrently* with the incremental update, resulting in true reactive planning. This feature represents a significant improvement over existing techniques for path planning. In addition, we have discussed the following useful properties of harmonic functions:

- No spurious ("local") minima. This implies that the effector will always reach a goal if the environment has been modeled well.

- Completeness up to discretization. If there is no path from the current configuration to a goal, this can be detected.

- Behavioral variability through superposition. This allows a variation between paths which graze obstacle surfaces, and paths which are repelled from obstacles. Such variation can be used to optimize performance.

- Robust control in the presence of environmental model error. Unexpected forces result in the "sidetracking" of the effector to alternative streamlines.

- Provides a framework for acquisition, extension, and refinement of environment models.

5.1 Future Work

The topics discussed in this paper provide the foundation for ongoing projects in the Laboratory for Perceptual Robotics at the University of Massachusetts:

- A Denning mobile robot platform is being used as a testbed for integrating real-time path planning with sonar sensing, visual navigation and machine learning techniques. The blending parameter k can be used to optimize performance with respect to different objective functions (e.g., path length, travel time). Q-learning techniques are currently being used for this purpose [28] .

- A harmonic function chip similar to that described by Tarassenko and Blake is in the prototype phase. It is anticipated that grids of modest size (100 by 100) will be able to compute harmonic functions within a few microseconds.
- Harmonic functions provide a model for planning in the mammalian central nervous system [29] . Recent results in neuroscience suggest that primary neurons of the striatum could be computing harmonic functions to drive movement. This theory is consistent with the pathology of certain movement disorders (Parkinson's and Huntington's Diseases) and with a variety of recent experiments in neurology and neuroscience (e.g., [30, 31]). The details of the theory appear in Connolly and Burns (1993)[29] .

The authors would like to acknowledge Brian Burns, Gerald Pocock, and Kamal Souccar for their useful insights, encouragement, and assistance. We would also like to thank Elizeth Giestal de Araujo, Jefferson Coelho and Manfred Huber for their assistance.

References

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

[2] 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.

[3] 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.

[4] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pages 500–505. IEEE, March 1985.

[5] Bruce H. Krogh. A generalized potential field approach to obstacle avoidance control. In *Robotics Research: The Next Five Years and Beyond.* Society of Manufacturing Engineers, August 1984.

[6] W. S. Newman and N. Hogan. High speed robot control and obstacle avoidance using dynamic potential functions. Technical Report TR-86-042, Philips Laboratories, November 1986.

[7] Damian Lyons. Tagged potential fields: An approach to specification of complex manipulator configurations. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, pages 1749–1754. IEEE, April 1986.

[8] Ronald C. Arkin. Towards cosmopolitan robots: Intelligent navigation in extended man-made environments. Technical Report 87-80, COINS Department, University of Massachusetts, September 1987.

[9] John K. Myers. Multiarm collision avoidance using the potential field approach. In *Space Station Automation*, pages 78–87. SPIE, 1985.

[10] Daniel E. Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, pages 1–6. IEEE, April 1987.

[11] Tomás Lozano-Pérez. Robotics (correspondent's report). *Artificial Intelligence*, 19(2):137–143, 1982.

[12] John F. Canny and Ming C. Lin. An opportunistic global path planner. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 1554–1559, May 1990.

[13] Jérôme Barraquand and Jean-Claude Latombe. Robot motion planning: A distributed representation approach. *International Journal of Robotics Research*, 10(6):628–649, December 1991.

[14] S. Akishita, S. Kawamura, and K. Hayashi. Laplace potential for moving obstacle avoidance and approach of a mobile robot. In *1990 Japan-USA Symposium on Flexible Automation, A Pacific Rim Conference*, pages 139–142, 1990.

[15] G. D. McCann and C. H. Wilts. Application of electric-analog computers to heat-transfer and fluid-flow problems. *Journal of Applied Mechanics*, 16(3):247–258, September 1949.

[16] John Francis Canny. *The Complexity of Robot Motion Planning.* PhD thesis, Massachusetts Institute of Technology, 1987.

[17] Tomás Lozano-Pérez. Automatic planning of manipulator transfer movements. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(10):681–698, October 1981.

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

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

[20] Stephen F. McCormick, editor. *Multigrid Methods*. Society for Industrial and Applied Mathematics, Philadelphia, 1987.

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

[22] 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.

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

[24] 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.

[25] Kenneth Salisbury, William Townsend, Brian Eberman, and David DiPietro. Preliminary design of a whole-arm manipulation system (WAMS). In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pages 254–260. IEEE, April 1988.

[26] Manfred Huber and Roderic A. Grupen. 2-d contact detection and localization using proprioceptive information. Technical Report 92-59, Computer Science Department, University of Massachusetts, August 1992.

[27] Edward Cheung and Vladimir Lumelsky. Motion planning for a whole-sensitive robot arm manipulator. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 344–349. IEEE, May 1990.

[28] S. P. Singh. Personal communication.

[29] Christopher I. Connolly and J. Brian Burns. A model for the functioning of the striatum. *Biological Cybernetics*, 68(6):535–544, 1993.

[30] John P. Walsh, Carlos Cepeda, Chester D. Hull, Robin S. Fisher, Michael S. Levine, and Nathaniel A. Buchwald. Dye-coupling in the neostriatum of the rat: II. decreased coupling between neurons during development. *Synapse*, 4:238–247, 1989.

[31] Garrett E. Alexander and Michael D. Crutcher. Functional architecture of basal ganglia circuits: neural substrates of parallel processing. *Trends in Neurosciences*, 13(7):266–271, 1990.

List of Figures

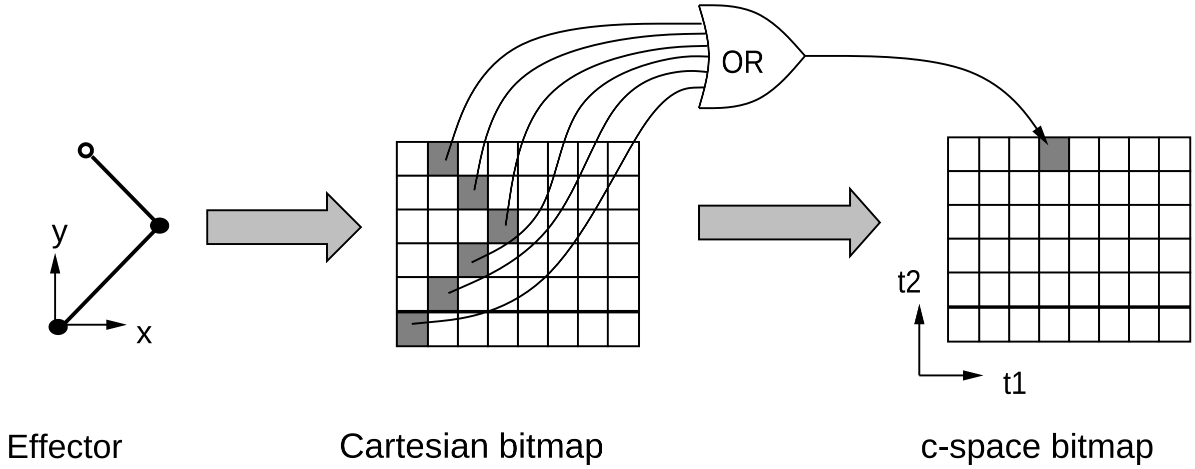

Figure 1: A schematic showing the conversion of a Cartesian bitmap to a C-space bitmap. If any portion of the manipulator's structure is inconsistent with known C-space boundaries, this manipulator configuration is marked in the C-space bitmap.

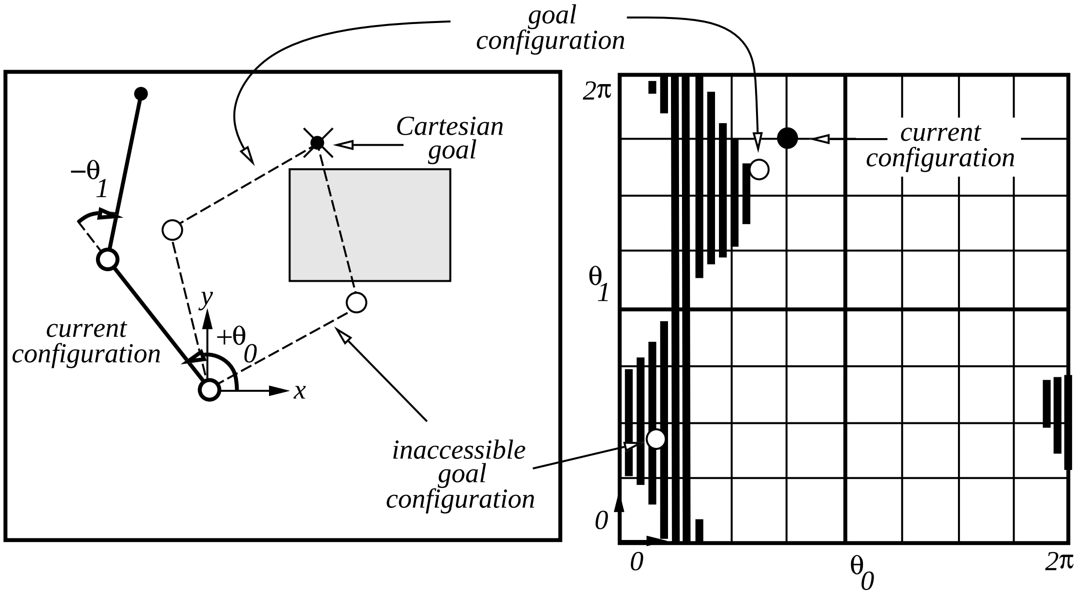

Figure 2: C-space for a 2-link arm.

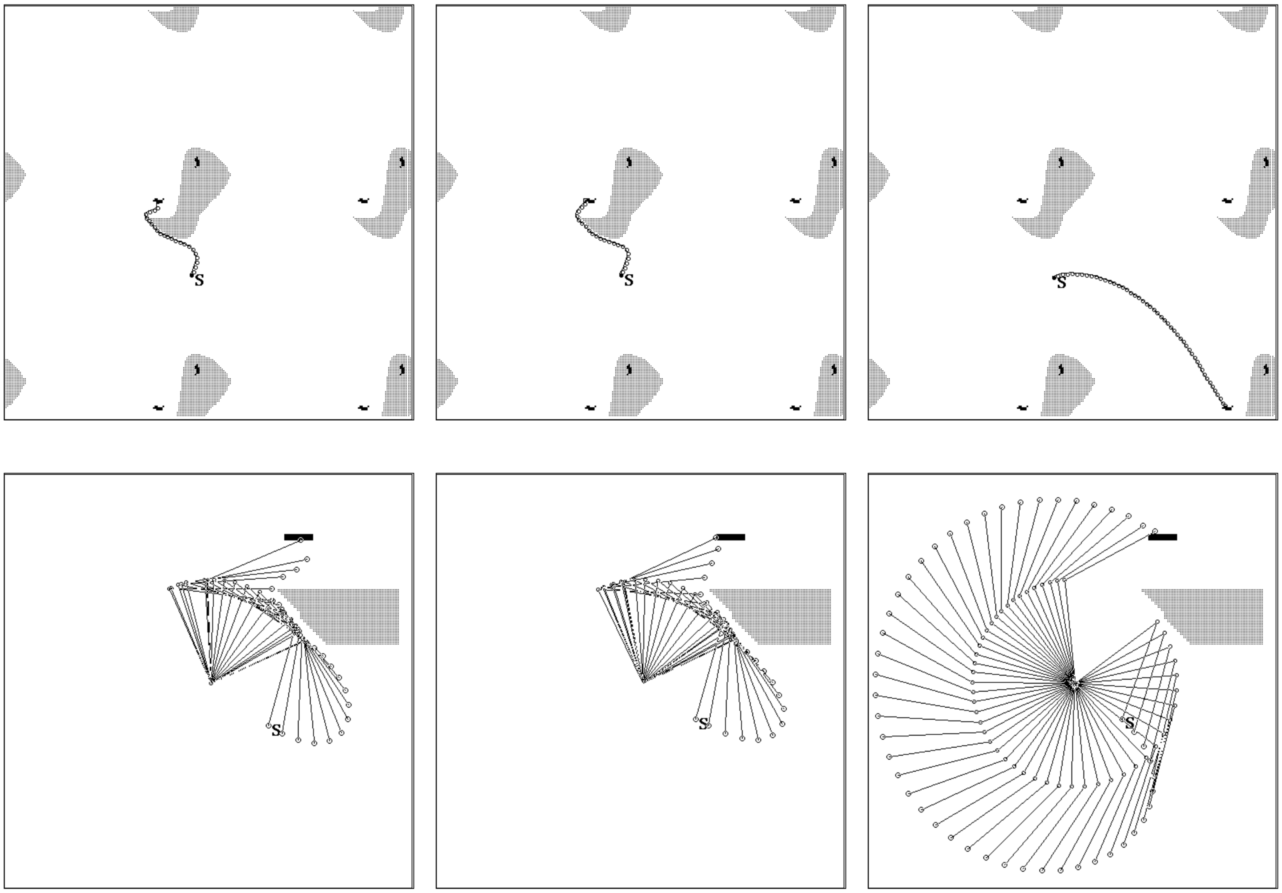

Figure 3: C-space (top row) and Cartesian (bottom row) paths for a 2-link arm. Values of k are 0.0 (left column), 0.5 (middle column), and 1.0 (right column). Changes in k can produce extreme changes in the path.

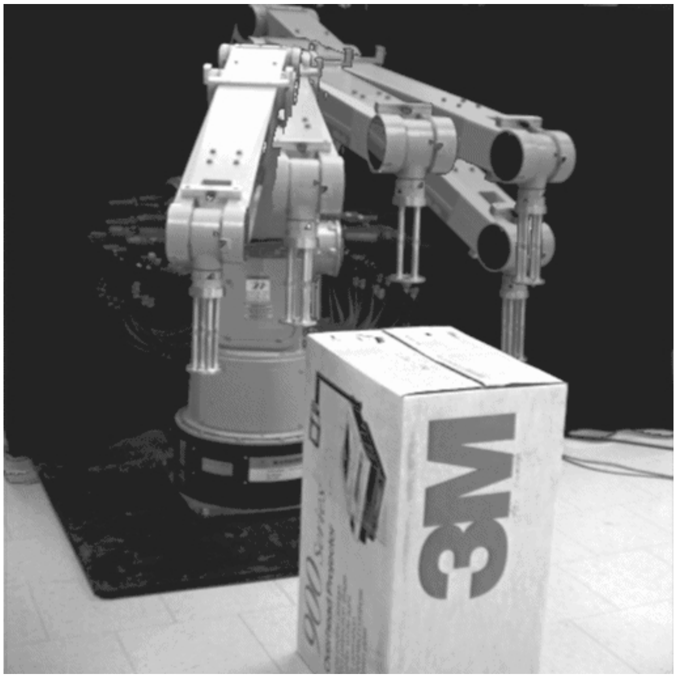

Figure 4: P-50 executing a three-dimensional path with a box as obstacle.

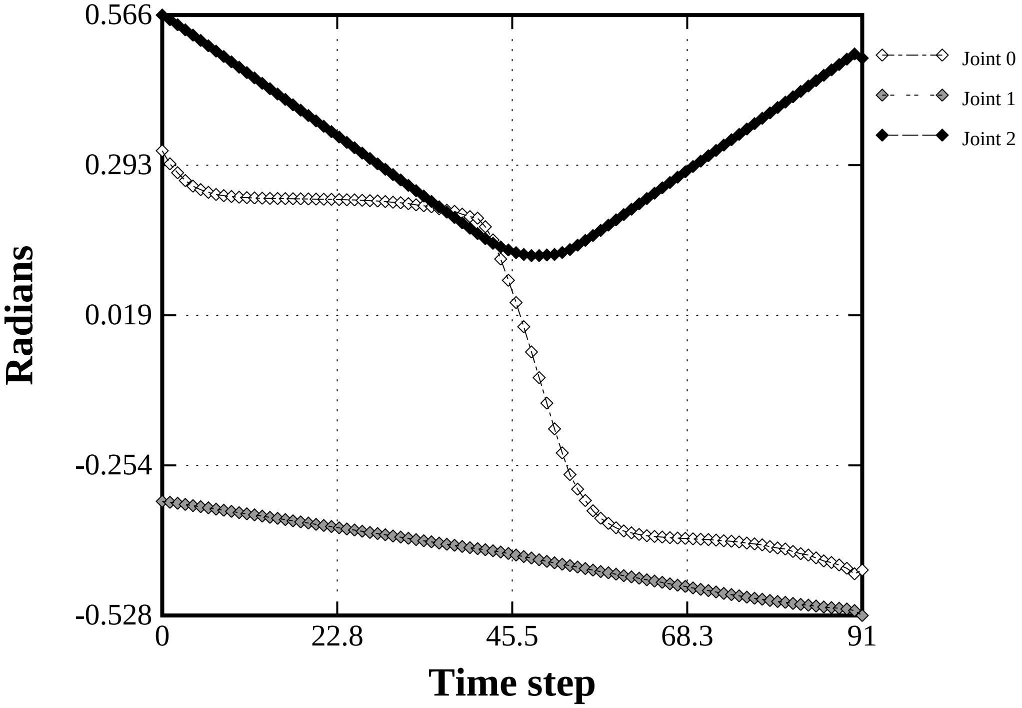

Figure 5: Joint angles as a function of time.

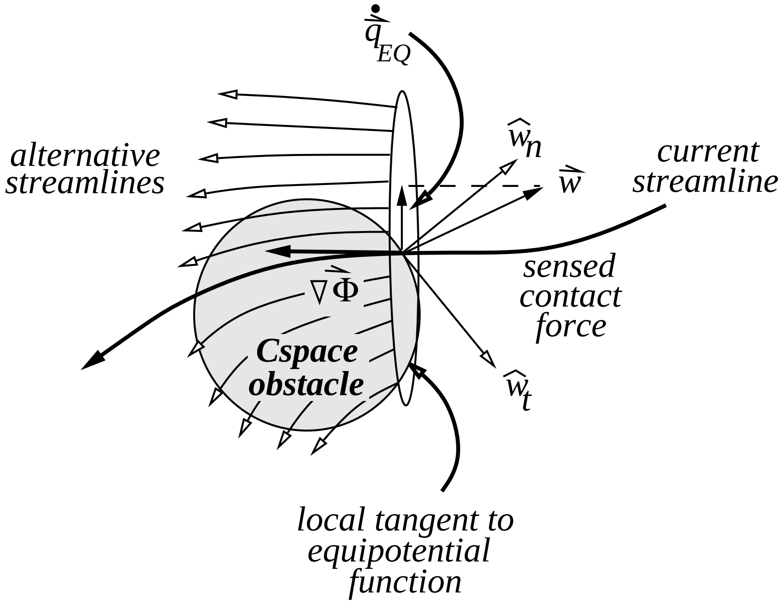

Figure 6: The harmonic function in the neighborhood for a contact with an unmodeled C-space obstacle.

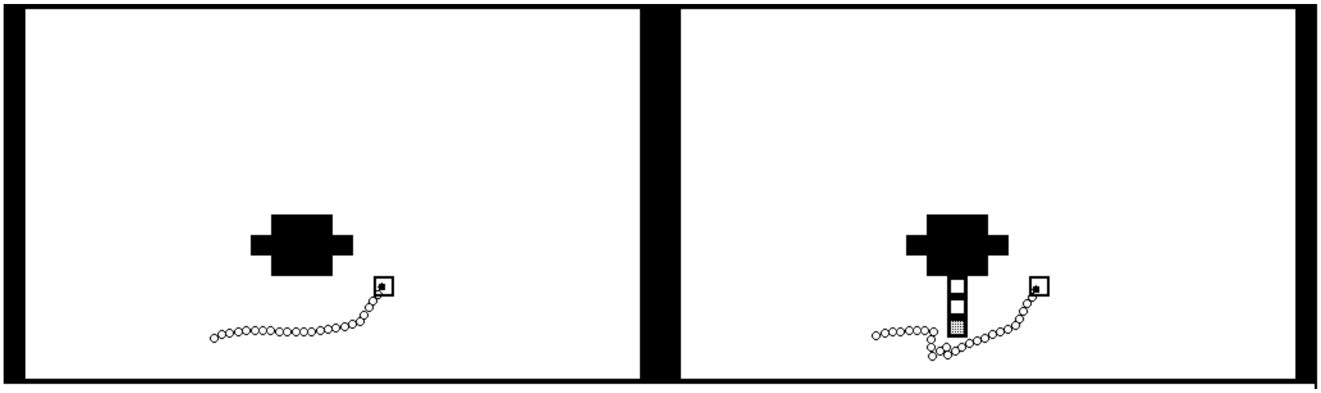

Figure 7: Example of a harmonic function computed over a 2-dimensional C-space (left) and using admittance control (right). Legend: small circle = effector; filled square = obstacle; large open square = goal; small open square = unsensed obstacle.

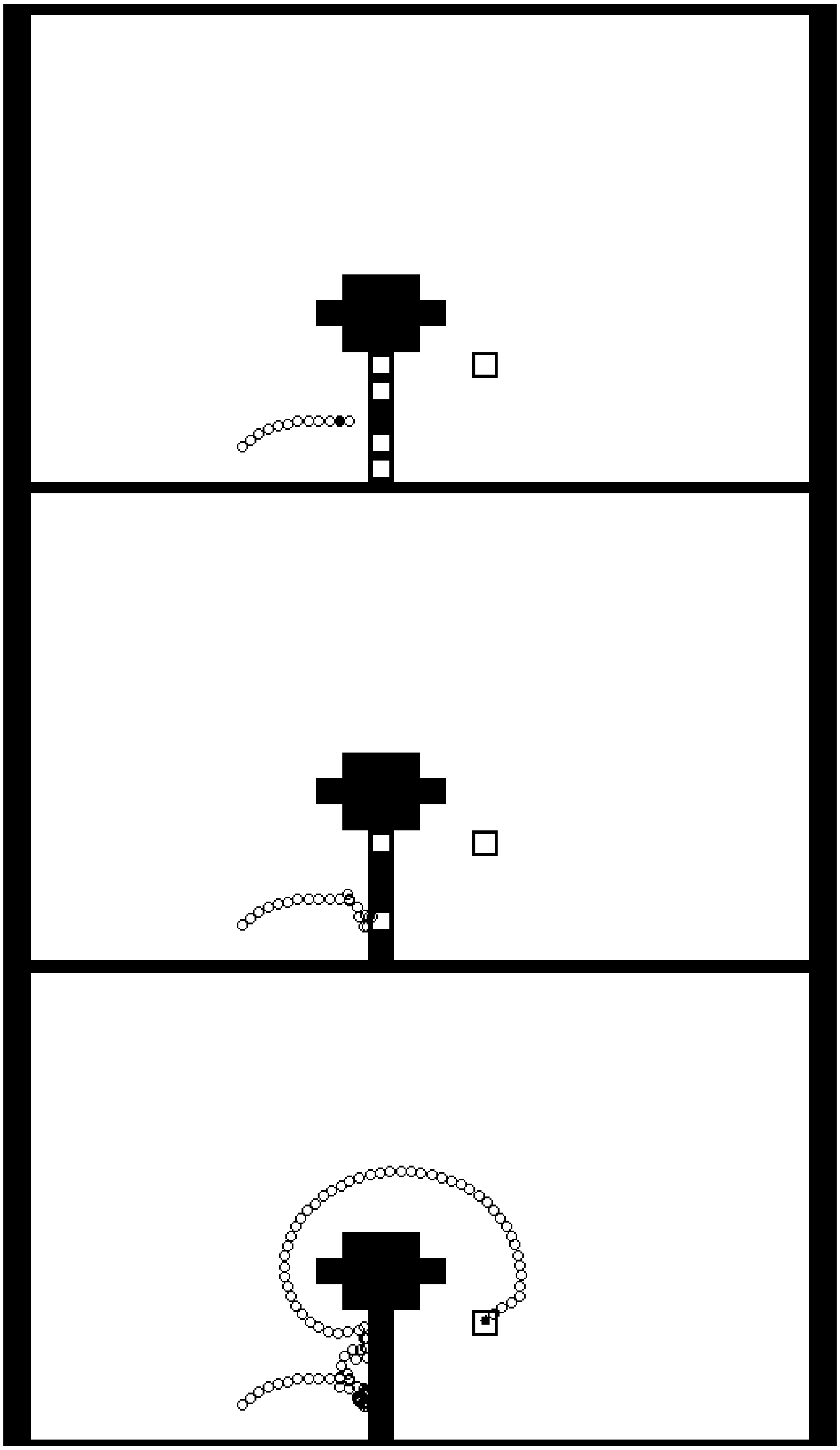

Figure 8: Three frames showing the combination of admittance with harmonic function computation.

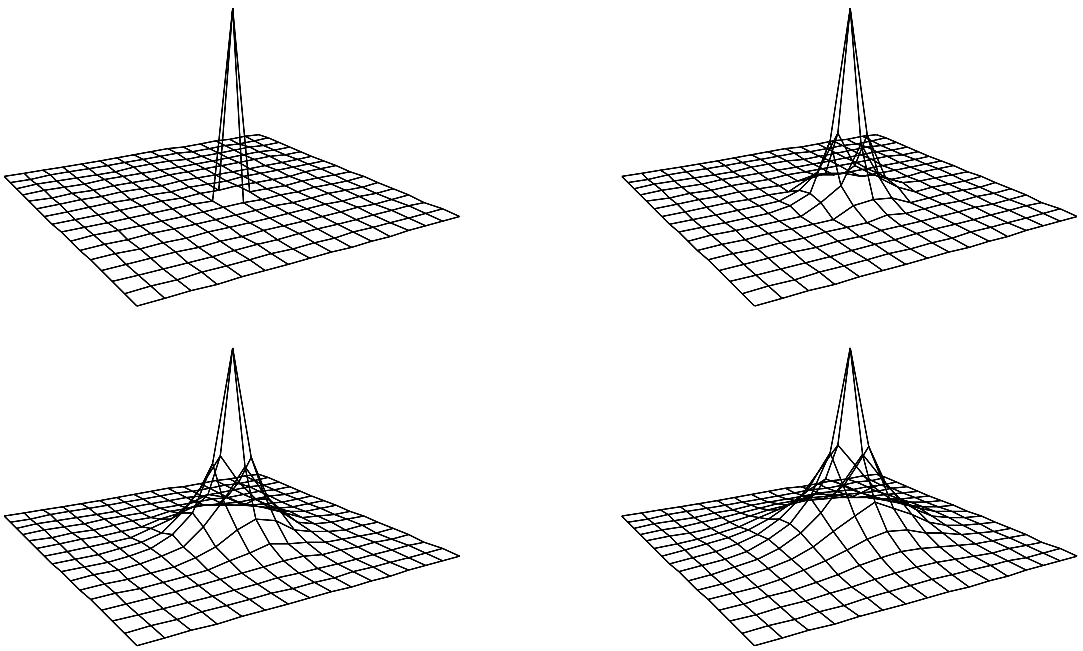

Figure 9: The impulse response of Jacobi computation of a harmonic function. This figure shows 4 Jacobi iterations left to right and top to bottom. The completely converged plot is shown in the lower right.