Nonholonomic Path Planning Using Harmonic Functions *

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

June 24, 1994

Abstract

A method is presented for planning obstacle-avoiding paths for a system which exhibits nonholonomic constraints. The method is based on the use of harmonic functions. Linear constraints on the velocity of a nonholonomic system can be directly expressed as Neumann boundary conditions for a harmonic function. Such boundary conditions are easily represented in a resistive network. The resulting potential represents an integration of nonholonomic constraints over an admissible subset of configuration space. The method is applied to path planning for simple wheeled vehicles.

1 Introduction

This paper draws on the relationship between harmonic functions and resistive networks to propose a new method for planning the motion of nonholonomic systems. Simple nonholonomic constraints [1] restrict the velocity of mechanical systems to a linear subspace of their configuration space. These constraints bear a resemblance to the Neumann boundary condition for harmonic functions. Although nonholonomic constraints are not globally integrable,

*UMass Amherst Computer Science Department technical report UM-CS-1994-050.
†This work is supported in part by the National Science Foundation under grants CDA-8922572, IRI-9116297 and IRI-9208920.

harmonic functions may be used to integrate these constraints over an admissible *subset* of configuration space, resulting in a potential which guides the vehicle to a goal configuration.

Harmonic functions have been used in planning obstacle-avoiding trajectories for holonomic systems (see Connolly and Grupen [2]). These functions are solutions to Laplace's equation and can be computed using resistive networks [3, 4]. Harmonic functions exhibit several properties which are useful in robotics applications. These properties and their applications are discussed more fully in [2], but include the following:

- At any point, the function is the (weighted) average of values in the neighborhood.
- The function exhibits no local minima, other than at the specified goals.
- Trajectories produced by this method are smooth almost everywhere.
- The function can be interpreted directly as a collision probability [5].
- The method is complete.

Methods for computing these functions are typically very simple [6]: obstacles and goals represent boundary conditions, and are held to fixed high and low potentials, respectively. A relaxation is then performed on the remaining (free) points, such that the function at any point is the (weighted) average of neighboring values. Other, more general properties of harmonic functions are discussed in detail in [7] and [8].

2 Nonholonomic Constraints

Nonholonomic systems have received some attention in the recent robotics literature [9, 10, 11, 12, 13, 14, 15, 16, 17]. This is a natural consequence of the interest in planning and controlling the motion of wheeled vehicles, since these are so common and are properly treated as nonholonomic systems. A nonholonomic mechanical system is one whose motion in configuration space is locally constrained. Traditional path planning techniques directly treat mechanical systems as being holonomic: it is sufficient to model such systems as freely moving points in a configuration space. A nonholonomic system can still be treated as a point in an appropriate configuration space, but this point cannot move freely. It is constrained to move instantaneously in a particular direction. One example of a nonholonomic system is a steerable wheel in contact with the ground. The wheel can roll forward or backward, but not to either side. In this paper, we show that harmonic functions can be employed to treat nonholonomic constraints directly. In general, it is possible to locally constrain the flow for a harmonic function in order to impose the nonholonomic constraints.

Kane [1] describes a holonomic system as one whose configuration variables are independent. In other words, the system can be treated as a point moving freely through configuration space. This picture is not sufficient to model many vehicular systems. In wheeled vehicles, for example, the contact between a wheel and the ground locally constrains the system to move in a particular direction. The wheel cannot translate sideways along the axle. Such a system has *nonholonomic* constraints on its configuration variables. Kane defines simple nonholonomic constraints on the configuration q as:

$$\sum_{i=1}^{n} A_{ij}\dot{q}_i + B_j = 0 \qquad (j = 1, \ldots, m) \tag{1}$$

where n is the dimension of the configuration space, and m is the number of constraints imposed. Equation 1 is a linear constraint on the configuration space velocity (or instantaneous rate of change) of the mechanical system. Simple nonholonomic constraints restrict the instantaneous velocity of the system to be within some linear subspace of configuration space. Figure 1 illustrates such a system.

3 Nonholonomic Planning

Some treatments of planning in nonholonomic systems are based on the notion that a suitable sequence of infinitesimal transformations can bring any simple nonholonomic system into any new configuration [11]. The theory behind this relies on the notion that a properly defined configuration space defines a Lie group, and that a suitable sequence of group operations

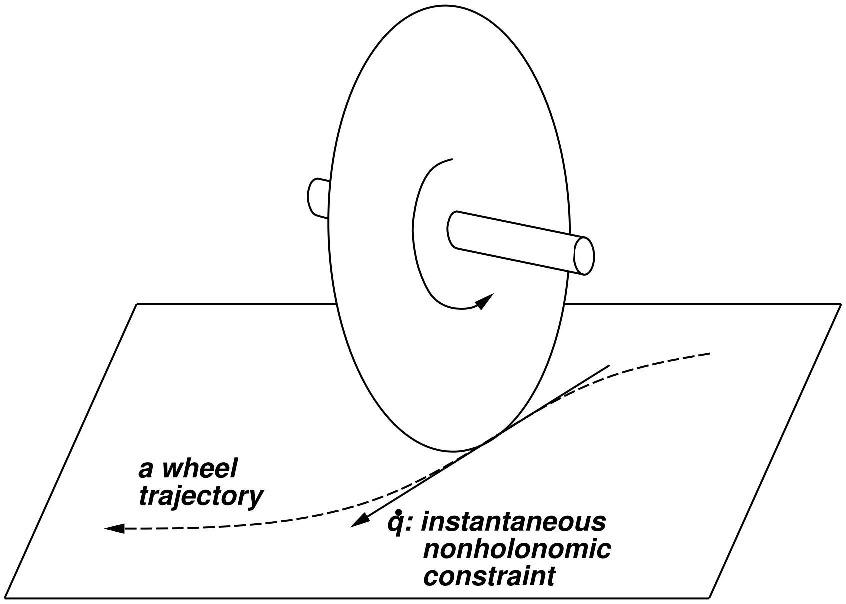

Figure 1: The rolling wheel: a simple nonholonomic system.

(Lie brackets) can be constructed by which the system can travel infinitesimally to reorient itself (see, for example, Li and Canny [11]). The intuitive picture here is approximately the same principle used to maneuver a car into a tight parking space. The common treatment of nonholonomic path planning involves the creation of a canonical path (using a conventional path planner) followed by an infinitesimal reorientation stage to properly align the system to this path, followed by a path execution [10, 11]. A final reorientation phase may be necessary to obtain a desired goal orientation. One drawback of this technique is that in practice, the reorientation phases can be time-consuming. In addition, physical systems cannot make truly infinitesimal movements. This places practical limits on the ability of a system to reorient itself arbitrarily.

Samuel and Keerthi [15], treat nonholonomic constraints directly as the boundary conditions for an optimal trajectory, where the optimality criterion is minimum x, y distance and minimum integrated steering angle. In contrast to other methods, the constraints in [15] are incorporated directly into the trajectory generation algorithm. However, the penalty function used for avoiding obstacles in [15] is not well specified, and may not result in a complete method. Moreover, this method appears to require the specification of intermediate goals. This is not required in the harmonic function approach. Barraquand and Latombe [9] use an exhaustive search to incorporate nonholonomic constraints. This, however, can be computationally expensive, especially in large cluttered environments. In contrast, harmonic functions which are computed using resistive networks can plan such trajectories rapidly. Their speed also allows such networks to respond to changing environments in a predictable fashion [2].

Nonholonomic constraints are equivalent to Neumann boundary conditions for the harmonic function. Neumann boundary conditions constrain the gradient of function ϕ:

$$\nabla\phi \cdot \vec{n} = d \tag{2}$$

where $\vec{n}$ is the normal to a surface which constrains the flow, and d is some constant. In [2], d was set to zero, implying a flow that is orthogonal to the normal vector. Recall the nonholonomic constraints in Equation 1. We will consider one constraint, written as:

$$\sum_{i=1}^{n} a_i \dot{q}_i + b = 0 \tag{3}$$

This can be written in vector form as:

$$\dot{\vec{q}} \cdot \vec{a} = -b \tag{4}$$

Normalizing the vector $\vec{a}$, and rewriting the resulting vector as $\vec{n}$ yields:

$$\dot{\vec{q}} \cdot \vec{n} = \frac{-b}{||\vec{a}||} \tag{5}$$

In harmonic function control, the gradient of the harmonic function ϕ is used to determine the tangent of the trajectory. Thus, the velocity of the system $\dot{\vec{q}}$ will be proportional to $\nabla\phi$. The constant c will denote the proportionality constant. This yields:

$$c\nabla\phi \cdot \vec{n} = \frac{-b}{||\vec{a}||} \tag{6}$$

$$\nabla\phi \cdot \vec{n} = \frac{-b}{c\|\vec{a}\|} \tag{7}$$

This is clearly a form of the general Neumann boundary condition. Thus, it is always possible to represent simple nonholonomic constraints as a Neumann boundary condition for the path planning problem. To see that simple systems always admit solutions (i.e., are not overconstrained), the resistive network formulation will be useful.

3.1 Resistive Networks

The intuitive explanation for the nonholonomic constraints for a wheel is that the wheel can only go in one direction for a particular steering angle. The wheel can be represented using a 3-dimensional configuration space: x, y, θ, where θ is the wheel's steering angle ($\theta \in [0, 2\pi)$). The wheel will also be allowed to move forward and backward, if necessary.

Harmonic functions can be computed using resistive networks. This can be seen by examining the discrete form for Laplace's equation defined over a grid:

$$u(x) = \frac{1}{2n} \sum_{i=1}^{2n} u(x_i) \tag{8}$$

where the x_i are the grid neighbors. The value at a point x, therefore, is simply the average of its neighbors' values. This equation is identical to Kirchhoff's voltage law for resistive networks. It is derived in [2], but can also be found in many other sources ([6] or [18], for example). The resistive grid can be used as follows: nodes corresponding to obstacle configurations are pulled high (1 *volt*), while goal nodes are pulled down (0 *volts*). Current will then flow between obstacle nodes and goal nodes, and the steepest voltage drops at free nodes correspond to a gradient descent of the harmonic function (voltage as a function of grid position). In practice, paths are generated by interpolating the function in the grid, and projecting the interpolated gradient onto the vehicle heading.

Consider a resistive grid constructed to represent a configuration space. For a holonomic system, the grid nodes are connected to their manhattan neighbors. The point representing the current configuration is allowed to move in any direction in the grid. Figure 2 shows the

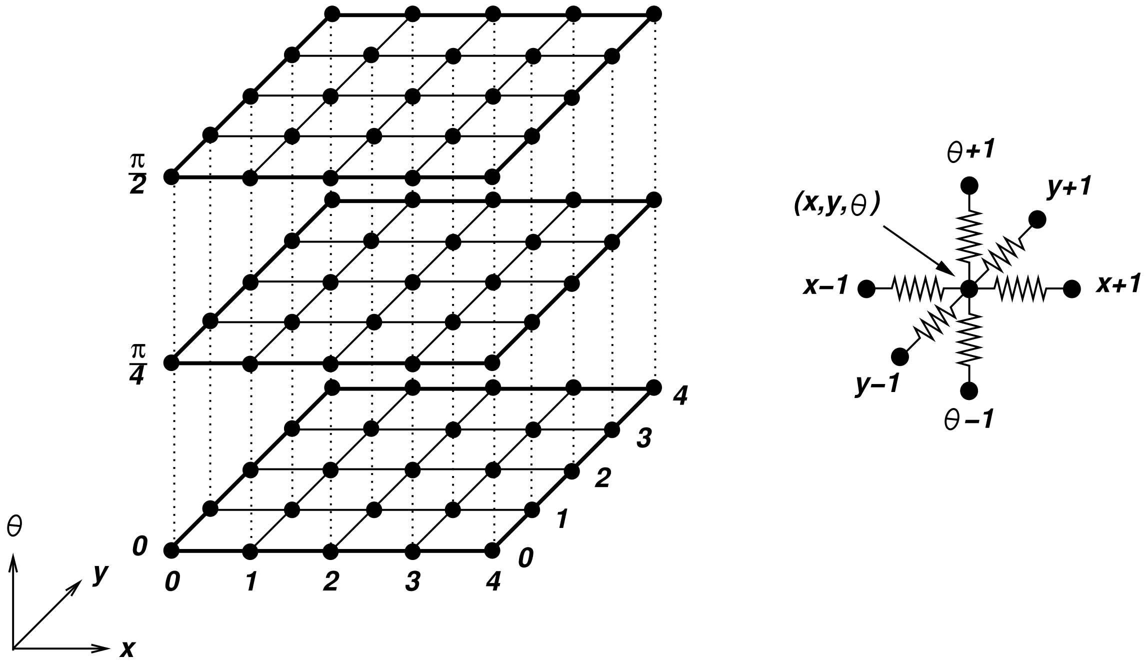

Figure 2: The holonomic 3D grid: lines denote resistances, and every interior grid node is connected to its manhattan neighbors.

connectivity of such a grid for a simple x, y, θ cspace. Only three orientations are shown, corresponding to $0, \frac{\pi}{4}$, and $\frac{\pi}{2}$. Since these steering angles constrain the system, we must design grid conductances to impose the appropriate nonholonomic constraints. With the θ discretization illustrated, this can be accomplished by connecting the appropriate subset of the 8 nearest neighbors in each of the x, y-planes. A grid connectivity that properly captures the nonholonomic constraints for each value of θ is illustrated in Figure 3. Note that in each x, y-plane, only those nodes that are neighbors in the direction of θ are connected. Note also that every node is connected to its neighbors in the adjacent constant-θ planes. Thus, the grid is still connected (that is, there is a path between any two points), but the legal paths through the grid are now constrained. Analogously, the valid flows for a nonholonomic system in the continuous case are constrained by the current value of θ. Since the nonholonomic grid is still connected, the Dirichlet problem is well-posed. Figure 3 also illustrates a trajectory computed on the grid from $(0, 0, 0)$ to $(4, 4, \frac{\pi}{2})$ in configuration space. Solution trajectories

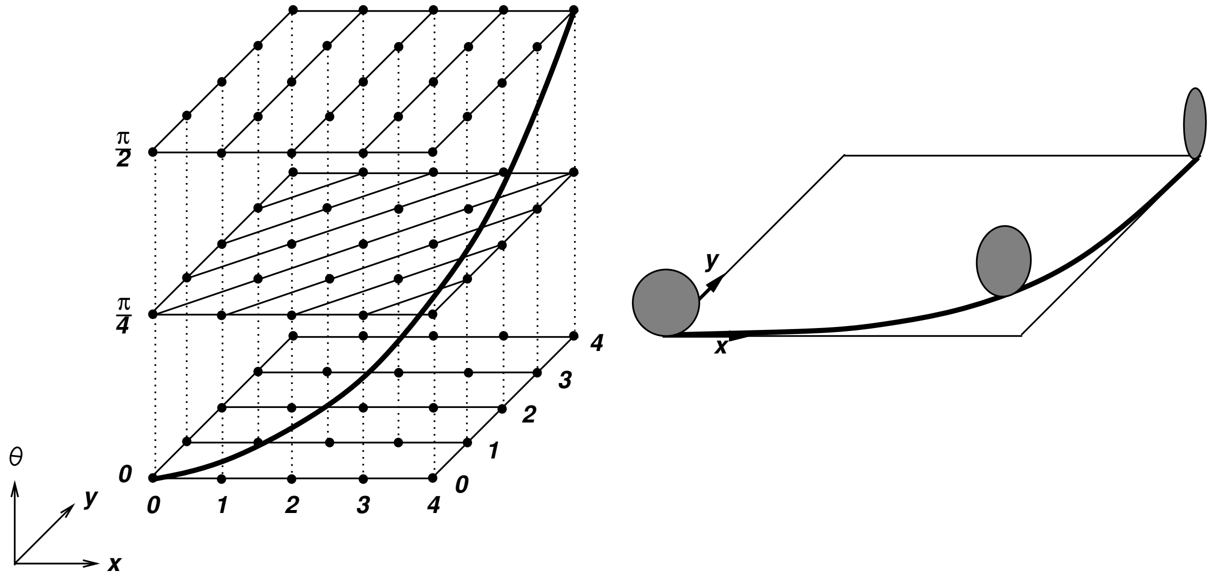

Figure 3: The nonholonomic 3D grid: each constant-θ plane has restricted connections in order to constrain the flow to be in the direction of θ.

will turn θ as the wheel is moving through x, y, until the wheel is perpendicular to its initial orientation. The bold curve is the trajectory of the system in cspace. On the right is a diagram showing the motion of a wheel corresponding to this trajectory. In contrast to more conventional nonholonomic planning methods, this technique allows continuous and simultaneous changes in the constraining degree of freedom (e.g., the steering angle), rather than a sequence of discrete infinitesimal reorientations.

4 Implementation

4.1 The Unicycle Controller

The examples in this section rely on a simulation of a resistive grid which is similar to the one shown in Figure 3. The system under consideration is a single wheel. Four θ directions ($\theta \in \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$) are explicitly represented in the grid. The θ dimension wraps around so that $\theta = \pi$ maps to the $\theta = 0$ plane. This results in a configuration space which represents all unique headings of the wheel. The wheel has no preferred forward direction.

The system of equations corresponding to the resistive grid is derived from Kirchhoff's law: the voltage at a node is the weighted average of neighboring voltages. In every constant-θ plane, each node is connected to two neighbors in θ (denoted by $\theta+$ and $\theta-$), and two x, y neighbors (denoted $xy+$ and $xy-$). The x, y neighbors are determined by the value of θ and the local 8-neighborhood in that plane. For example, for a node at (x, y) in the $\theta = \frac{\pi}{4}$ plane, the $xy+$ neighbor has coordinates $(x+1, y+1)$, and the $xy-$ neighbor is at $(x-1, y-1)$, likewise for the other constant-θ planes. The resulting system of equations can be written as:

$$\phi(x, y, \theta) = \frac{1}{4}(\phi_{xy+} + \phi_{xy-} + \phi_{\theta+} + \phi_{\theta-}) \tag{9}$$

In this case, inter-node conductances are all equal. This equation can be generalized to the case where inter-node conductances vary to generate a family of qualitatively distinct paths.

In implementation, the wheel's simulated translation is constrained to be in the direction of the current heading. The harmonic function is computed over the grid according to Equation 9, and trajectories are obtained by using the gradient at each configuration as a control command. The x, y component of the gradient (denoted by $\delta x, \delta y$) is projected onto the current heading to generate the appropriate displacement along the current heading:

$$\delta s = -(\delta x \cos\theta + \delta y \sin\theta),$$

this ensures that the interpolated harmonic function subscribes explicitly to the nonholonomic constraint. In each control cycle, the simulated wheel advances along θ by δs and is simultaneously rotated by the θ component of the gradient, $\delta\theta$.

Figure 4 shows an example of a trajectory planned using the resistive grid formulation. The wheel starts at an angle $\theta = 0$ (horizontal in the figure), and follows a smooth trajectory until the goal is reached at $\theta = \frac{\pi}{2}$.

In Figure 5, a second example is shown where the wheel must make its way out of a narrow channel, and must reach the goal in a horizontal orientation. In this case, the wheel stops and reverses direction after leaving the channel. The "cusp" (leftmost point) in the trajectory is caused by a gradient whose x, y components are nearly zero with respect to the

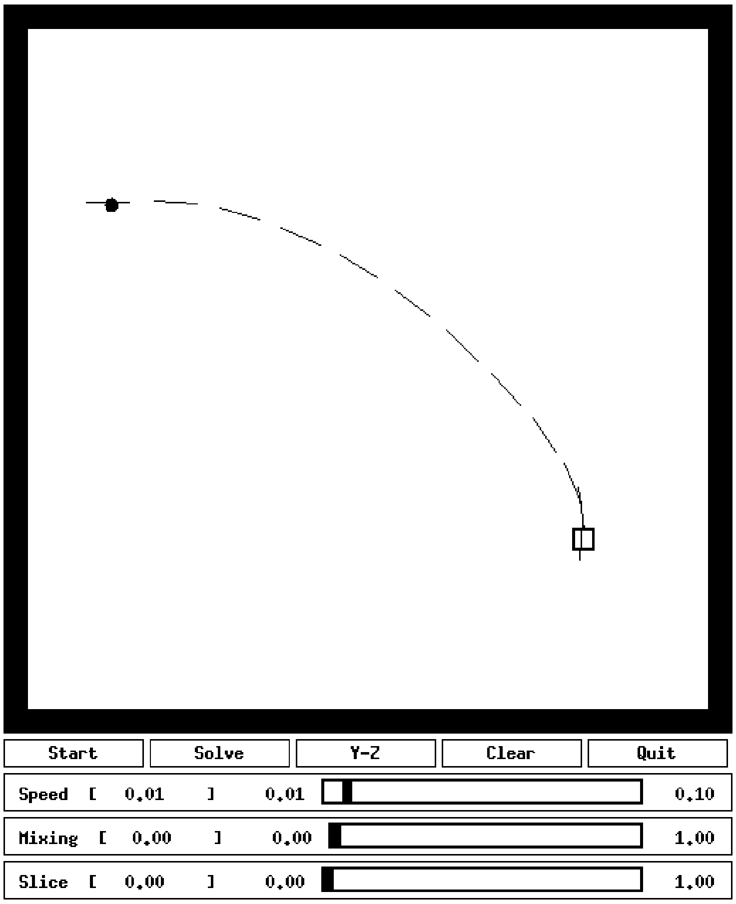

Figure 4: Example of a planned nonholonomic trajectory: the open square is a goal, the black circle indicates starting position. The wheel is displayed as a line drawn at the appropriate angle.

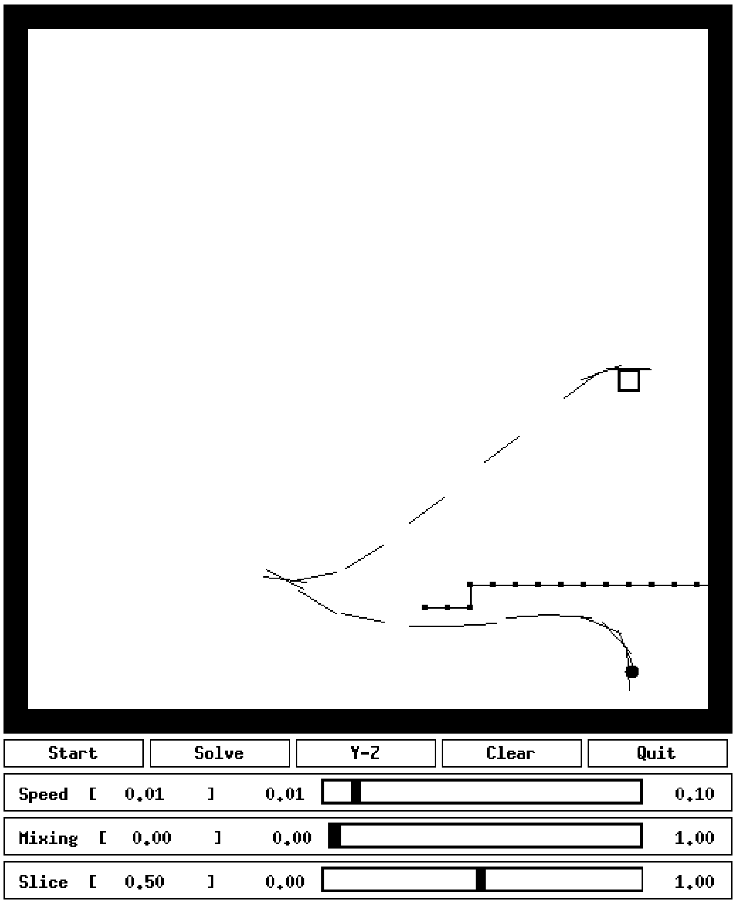

Figure 5: Example of another planned nonholonomic trajectory. Here, the wheel starts in the vertical orientation, and must reach the goal in a horizontal orientation. Note the point on the left, where the direction of travel is reversed.

current orientation. A slight change in θ (to orient the wheel toward the goal) increases the x, y gradient once again, resulting in a reversal of the wheel's previous direction of movement.

Finally, Figure 6 shows a more complicated set of channels. The start configuration is at $\theta = \frac{3\pi}{4}$, and the goal orientation is $\theta = \frac{\pi}{4}$. The wheel must exit the channel on the right, and reenter on the left. Again, there is a cusp in the trajectory where the x, y component crosses zero.

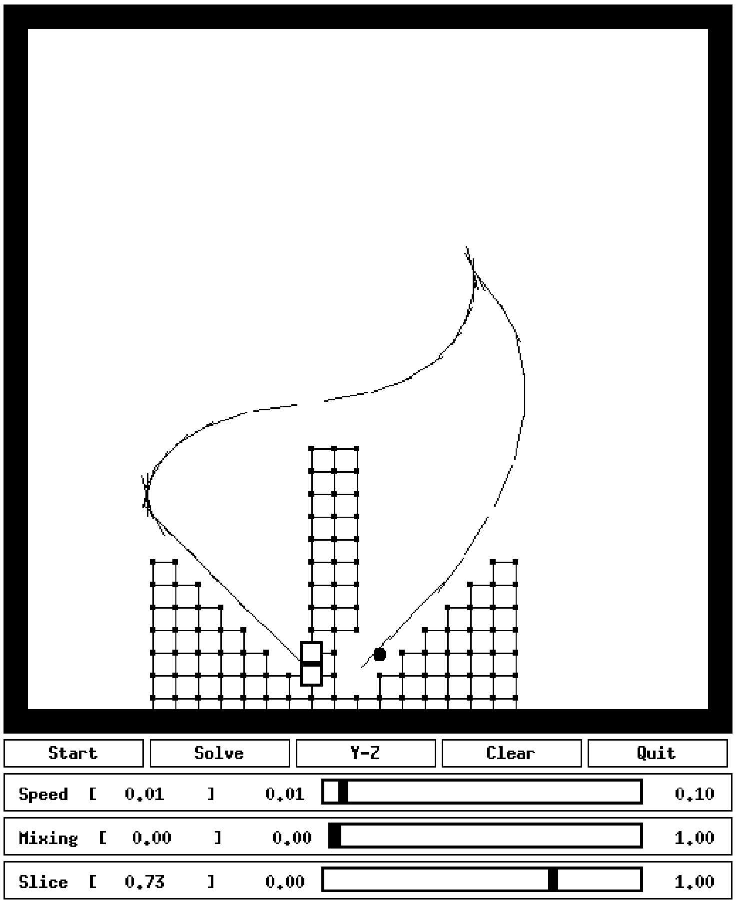

Figure 6: Example of a more complicated nonholonomic trajectory. The wheel must rotate 90 degrees to reach the goal in the left half of the Y-shaped corridor.

4.2 The Bicycle Controller

The examples in this section employ a four dimensional resistive grid spanning θ (absolute steering angle), ϕ (the absolute heading of the rear wheel), x, and y. The figures illustrate the front wheel of the bicycle as the arrow (to define absolute heading) with a circle to designated the (x, y) position that being controlled[1].

Eight unique θ and ϕ directions $\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{6\pi}{4}, \frac{7\pi}{4}\}$ are explicitly represented in the grid. The rotational configuration variables wrap around. This results in a configuration space which represents all unique headings of both wheels. The resulting grid has dimension $8 \times 8 \times 16 \times 16$. Unlike the previous example, here we do not express the nonholonomic constraints explicitly in terms of inter-nodal conductivities. In the unicycle examples, the wheel heading, θ, defined the flow constraints in the x, y-plane, namely, $(cos(\theta), sin(\theta))$. Since the constraint depends solely on θ, a grid aligned to $\pi/4$ increments can be explicitly represented in the 8-neighborhood of each x, y-node. The nonholonomic constraint for the bicycle is defined in the ϕ, x, y-plane, namely, $(\frac{sin(\theta-\phi)}{L}, cos(\theta), sin(\theta))$, where L is the distance between the axles of the two wheels. It is not an easy matter to align this constraint with the configuration grid. Instead, we used local support in the grid to interpolate the potential along the nonholonomic constraint and execute the same numerical relaxation (Equation 9) as before.

Figure 7 is a simple task that demonstrates how flow in the grid expresses the non-holonomic constraint. The square is the goal region in the configuration space. All four configuration variables are specified in the goal configuration. Often, however, it is sufficient to achieve the ϕ, x, y goal and to ignore the steering angle.

Figure 8 shows a slightly more complicated trajectory. Here, vehicle backs up from the initial horizontal, right facing configuration all the way to the ϕ, x, y goal where the steering angle is adjusted to achieve the goal. In the process of backing up, the rear wheel flips its heading relative to the steering wheel. The reason that this occurs is because there are no

[1]The circle is drawn at the point at which the steering wheel contacts the ground, and is the also the point at which the bicycle is articulated.

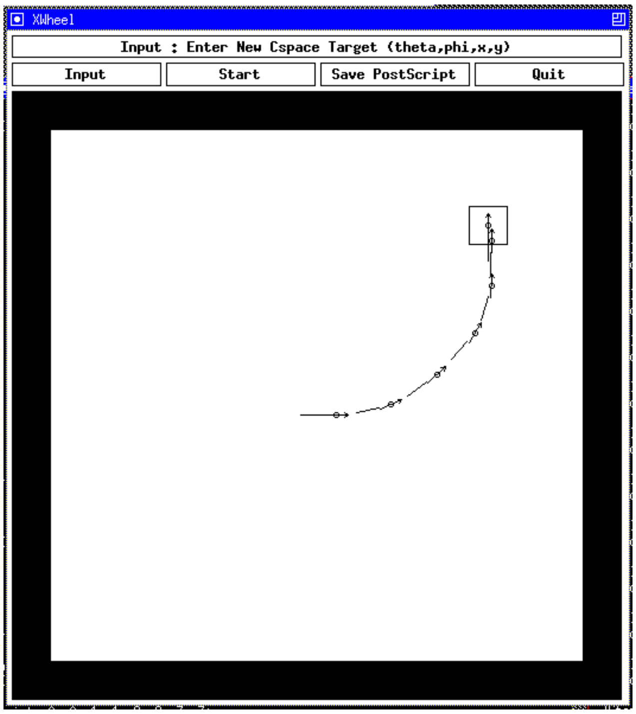

Figure 7: The four dimensional, nonholonomic bicycle: the task is to navigate from the horizontal configuration in the middle of the pane to the vertical configuration in the upper right corner.

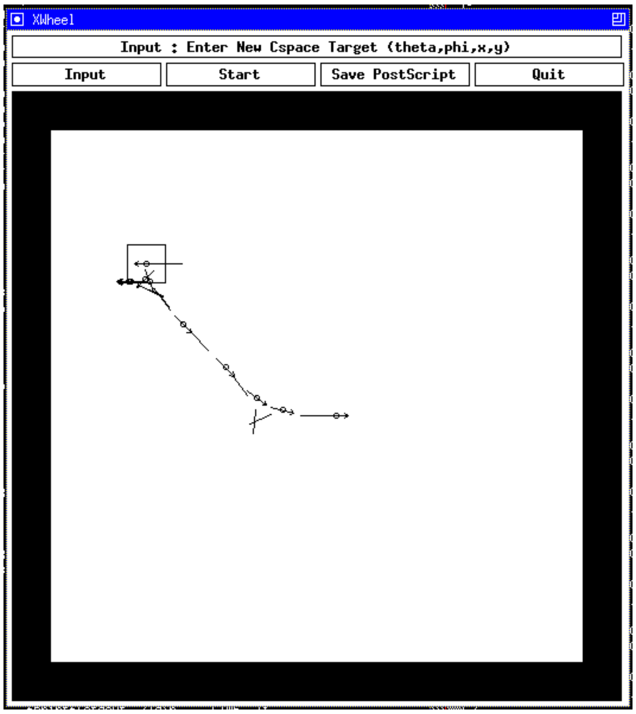

Figure 8: Example of a linear, nonholonomic bicycle trajectory from the middle, rightfacing configuration to the upper left, leftfacing configuration. The steering angle is permitted to vary between $-\pi$ and $+\pi$ relative to the rear wheel heading.

constraints on the relative steering angle in this example. This is a linear, nonholonomic system.

Figure 9 executes the same task as illustrated in Figure 8 but constrains the relative steering angle to remain within $\pm\frac{\pi}{2}$. The vehicle begins from the right facing horizontal configuration and drives forward while turning right toward the lower right corner of the pane. After a small forward excursion, the bicycle backs up along a straight path segment as shown. It repeats this strategy, small forward right turn followed by a straight backward path segment, two more times until its heading is oriented toward the goal. At this point, the bicycle executes a smooth forward path to the goal. This example highlights some of the ways in which the resistive grid can be tuned to generate different responses. First, the straight segments are not strictly continuous backward motions. They actually consist of small backward then forward displacements that direct the rear wheel in an approximately straight line. This is due to the inherent instability of this vehicle in the backward direction, but is unnecessary. This character of the path is affected by the magnitude of the step executed along the streamline, and the relative conductance in the θ direction versus the x, y, ϕ direction. Second, the constraint on the relative steering angle is expressed as a Dirichlet boundary condition. This is a *repulsive* boundary so that the steering angle tends to remain at a zero relative heading with respect to the rear wheel. This generates the straight line segments in the figure. This approach is a reasonable strategy when the optimality criterion is the minimum integrated steering angle[15], but is not suited to paths in which the number of *cusps* in the path is minimized[9]. Neumann boundary conditions are preferable in this case.

5 Summary

A method has been presented for incorporating nonholonomic constraints into harmonic function path planning. The key to this treatment is that the harmonic function approach is capable – through the application of Neumann boundary conditions – of constraining the system to move along a particular linear subspace of configuration space. The flow

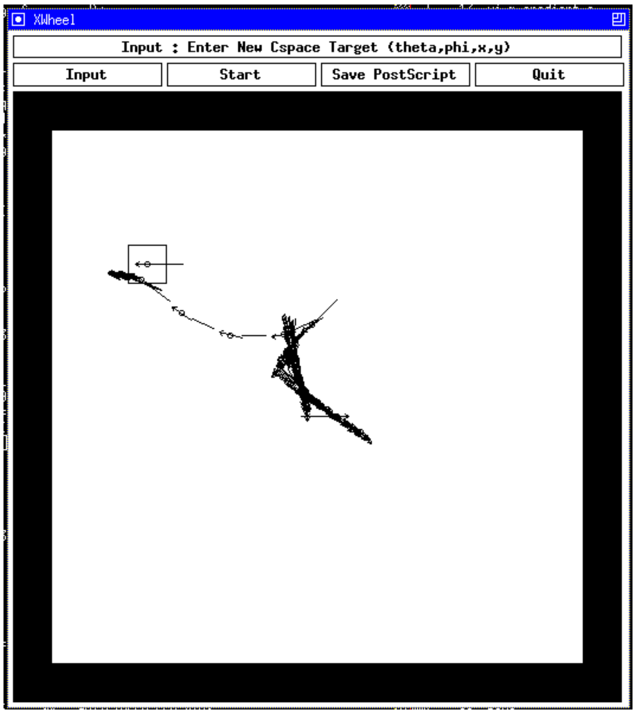

Figure 9: Example of a nonlinear, nonholonomic bicycle trajectory from the middle, rightfacing configuration to the upper left, leftfacing configuration. A Dirichlet boundary condition constrains the steering angle to $\pm\frac{\pi}{2}$ relative to the rear wheel heading.

corresponding to the harmonic function is constrained to a particular direction corresponding to the steering angle. In practice, it is fairly simple to implement this scheme using a constrained resistive grid.

In contrast to other approaches, the nonholonomic constraints are incorporated directly into the harmonic function path planning method. Of particular interest is the fact that here, changes in steering angle are made while the vehicle is in motion. Nonholonomic motion planners which require initial reorientation (e.g, [10, 11]) do not exhibit this characteristic. The harmonic function approach can vary the steering angle smoothly while the vehicle is in motion.

While the resistive grid is "safe", in the sense that obstacles will always be avoided, it is also flexible, and through the variation of grid resistances will allow a wide range of possible solutions to be expressed. The use of a resistive grid formulation begs the question: how should grid conductances be set to produce "optimal" paths? This question underscores one potentially powerful role for learning techniques in the context of robot motion planning. In general, it will be difficult to preset conductances for all possible environments or robot effectors. In the future, we intend to study ways in which learning algorithms can adaptively improve the path quality while compensating for robot dynamics by varying the conductances in the resistive grid.

References

[1] Thomas R. Kane. *Dynamics*. Holt, Rinehart and Winston, Inc., New York, 1968.

[2] Christopher I. Connolly and Roderic A. Grupen. The applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–946, October 1993.

[3] 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.

[4] 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.

[5] C. I. Connolly. Harmonic functions and collision probabilities. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 3015–3019. IEEE, May 1994.

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

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

[8] Sheldon Axler, Paul Bourdon, and Wade Ramey. *Harmonic Function Theory*, volume 137 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1991.

[9] Jérôme Barraquand and Jean-Claude Latombe. On nonholonomic mobile robots and optimal maneuvering. In *Proceedings of the IEEE International Symposium on Intelligent Control*, pages 340–347, September 1989.

[10] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pages 2–7, May 1989.

[11] Z. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Transactions on Robotics and Automation*, 6(1):62–72, February 1990.

[12] Th. Fraichard and C. Laugier. On line reactive planning for a non holonomic mobile in a dynamic world. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 432–437, April 1991.

[13] B. d'Andrea Novel, G. Bastin, and G. Campion. Dynamic feedback linearization of nonholonomic wheeled mobile robots. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2527–2532, May 1992.

[14] Brian Mirtich and John Canny. Using skeletons for nonholonomic path planning among obstacles. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2533–2540, May 1992.

[15] Sudhaker Samuel and S. Sathiya Keerthi. Numerical determination of optimal nonholonomic paths in the presence of obstacles. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, volume 1, pages 826–831. IEEE, May 1993.

[16] Zvi Shiller, William Serate, and Minh Hua. Trajectory planning of tracked vehicles. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, volume 3, pages 796–801. IEEE, May 1993.

[17] Ranjan Mukherjee and David P. Anderson. Nonholonomic motion planning using stokes' theorem. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, pages 802–809. IEEE, May 1993.

[18] Peter Doyle and J. Laurie Snell. *Random Walks and Electric Networks.* Carus Monographs in Mathematics. American Mathematical Society, 1984.