ICAR '97
Monterey, CA, July 7-9, 1997

A Wall-Following Method for Escaping Local Minima in Potential Field Based Motion Planning

Xiaoping Yun and Ko-Cheng Tan

Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943
yun@nps.navy.mil

Abstract

A wall-following method for escaping local minima encountered by the potential field based motion planning method used in real-time obstacle avoidance is presented. The new algorithm switches to a wall-following control mode when the robot falls into a local minimum. It switches back to the potential field guided control mode when a certain condition is met. A simple switch condition derived from monitoring the distance from the robot's position to the goal position is shown to be effective in escaping local minima in typical laboratory environments. A provision is built into the algorithm, allowing the robot to follow a wall in a different direction if the first attempt fails. The new algorithm is implemented on a Nomad 200 mobile robot. Simulation and experimental results are presented to demonstrate the usefulness of the method.

Keywords: Mobile robot, motion planning, potential field, and local minimum.

1 Introduction

Many motion planning methods have been developed for controlling autonomous mobile robots in the past decade [1, 2, 3, 4, 5, 6]. The potential field based method is one of them. The basic concept of the potential field method is to fill the robot's workspace with an artificial potential field in which the robot is repulsed from obstacles and is attracted to its goal position [7]. The method can be used for off-line global planning when the robot's environment is known a priori, or for real-time local planning when the environment is unknown and the presence of obstacles is measured by onboard sensors. This method is conceptually simple, easy to implement, and widely used in the community. However, it has the so-called local minimum problem.

Under the influence of an artificial potential field, the mobile robot moves in the negative gradient direction (or the steepest descent direction) from a high potential area to a low potential area. A properly constructed artificial potential field achieves its global minimum at the goal position where the gradient vector is zero. As expected, the mobile robot stops moving after reaching the goal position. But an artificial potential field may have other local minima where the gradient vector is zero as well. Consequently, the mobile robot may be trapped in a local minimum.

There are various efforts aimed at overcoming the local minimum problem. Those efforts fall into two categories: (i) establishing artificial potential fields that do not have local minima other than at the goal position; (ii) developing methods for escaping from local minima. Previous work in the first category includes the generalized potential field method [8], the navigation function method [2, 9], the vortex field method [10], and the harmonic function method [11]. The generalized potential field method [8] takes into consideration the robot's position as well as velocity. If the robot moves parallel to boundaries of an obstacle, the obstacle does not generate any repulsive forces to the robot. The navigation function introduced in [2, 9] is a special class of potential field functions that do not have any local minima other than at the goal position. In general, a global navigation function may not exist. But for a sphere world (a disc workspace with smaller disjoint disc-shaped obstacles) or any worlds that can be transformed into such a sphere world, it was shown that a navigation function can be constructed [2, 9]. Nevertheless, the constructive method requires a priori knowledge of obstacles, which makes it difficult to use the method as a real-time planner where obstacles are measured by onboard sensors. The vortex field method replaces the repulsive force field by the vortex velocity field of a liquid in the presence of obstacles [10]. A harmonic function is a solution of Laplace's equation. Solving Laplace's equation with obstacles' boundaries as boundary conditions results in a harmonic function that does not have local minima, which can be used as a potential function for motion planning [11]. Solving Laplace's equations requires a priori knowledge of obstacles, which cannot be used for real-time planning. It is also computationally expensive. Elliptical potential [12] is a method that does not completely eliminate local minima but reduces their number when obstacles are in simple convex shape.

Previous studies reported in [13, 14, 15, 5, 16] belong to the second category. The random walk method proposed in [13] enables the robot to escape from a local minimum by letting it move in a random direction for a random length of time and then switching back to the potential field guided motion again. Of course, the robot may need to walk in many directions before escaping from a local minimum. The convergence time of the random walk is studied in [14] based on Markov chain and diffusion processes. The best-first method is another method to escape from local minima [4]. This method gets the robot out by filling up the local minimum. It is more appropriate for off-line planning with a priori knowledge of the environment. If used for real-time planning, this method is less efficient since the robot will move around for a long time to fill up the entire valley. More recently, a multi-potential field based method was suggested [17]. The method employs multiple linearly independent potential fields that have the same global minimum at the goal position. When the robot is trapped in a local minimum using one potential field, it switches to next potential field in a circular fashion. Success using this method depends on a choice of potential fields that do not share the same local minima. The possibility that the robot will move in a loop may be avoided by using a random switching pattern. The heuristic search algorithm [15] recovers from a local minimum by selecting a set of via points. The closest to the present work is an algorithm reported in [5] which uses a wall-following method to recover from a local minimum, and the TangentBug algorithm presented in [16]. The algorithm reported in [5] uses the absolute difference between the robot-to-goal direction and the actual robot direction as a criterion to determine if the robot is in a local minimum and to switch from the wall-following to the potential field method. The algorithm may generate false local minima and result in indefinite loops even in very simple environment [5, 15]. In the TangentBug algorithm [16], a robot builds a local tangent graph based on its range data, and plans its motion using the local tangent graph. The algorithm provides global convergence. The construction of a local tangent graph requires detecting end points such as vertices of a polygon obstacle, which can be difficult using sonar range sensors.

In this paper, we describe a method for escaping local minima in a typical laboratory environment. When the robot falls into a local minimum, it switches from the potential field guided control mode to a wall-following control mode. In the wall-following control mode, the robot follows the contour of obstacles. It switches back to the potential field guided control mode when a certain condition is met. A simple switch condition is proposed in this paper and its effectiveness is validated by simulations and experiments. For certain environments, it is observed that the robot may return to the same local minimum when the robot follows a wall in a particular direction. The algorithm allows the robot to make a second attempt by following the wall in the opposite direction. With two attempts, the algorithm makes it possible to escape local minima in most laboratory environments. It should pointed out that the main contribution of this method is in its simplicity and ease for practical

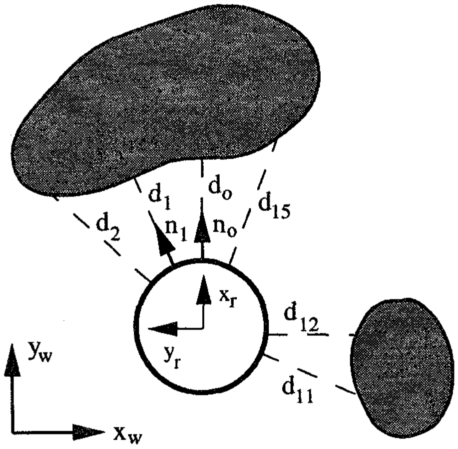

Figure 1: Illustration of coordinate systems and sonar measurements.

implimentation. It does not offer a complete solution to the local minimum problem. It may fail to overcome the local minimum problem in some cases. An example of such a case is presented in the paper.

2 Potential Field Based Motion Planning

In this section, we describe the potential field based motion planning that is implemented for simulation and experiment in this study. The attractive and repulsive potentials from [4] are adopted with some modifications. Modifications are made to accommodate the sensor systems of a Nomad 200 mobile robot which is our platform for simulation and experiment. Although the Nomad 200 mobile robot may be equipped with sonar, infrared, laser range finder, vision and other sensors, we will only use its sonar sensors in this study. The Nomad 200 is of cylindric shape and has a ring of 16 equally spaced sonar sensors. The beams of two adjacent sonar sensors are separated by 22.5 degrees. The 16 sonars are numbered sonar 0, sonar 1, ..., sonar 15 in counterclockwise direction.

To facilitate the description of potential fields, we use two coordinate systems as shown in Figure 1. x_w-y_w is an earth-fixed world coordinate system. x_r-y_r is a robot-fixed coordinate system. x_r is defined to be in the forward direction of the robot. It is also the pointing direction of the beam of sonar 0. The robot position in the world coordinate system is denoted by $q = [x \quad y]^T$. Orientation of the robot is measured by an angle θ from x_w to x_r.

The attractive potential takes the following form:

$$U_{att} = \begin{cases} \frac{1}{2}\xi \parallel q - q_{goal} \parallel^2, & \text{if } \parallel q - q_{goal} \parallel \leq \rho \\ \xi\rho \parallel q - q_{goal} \parallel, & \text{if } \parallel q - q_{goal} \parallel > \rho \end{cases} \tag{1}$$

where q_{goal} is the goal position, ρ is a constant distance, and ξ is a constant coefficient. When the robot approaches closer to the goal position, the attractive potential is parabolic in shape. It is conic in shape when the robot is at least ρ distance away from the goal position. The corresponding attractive force (the negative gradient of the attractive potential) in the world coordinate system is given by:

$$
\begin{aligned}
{}^{w}F_{att} &= -\frac{\partial U_{att}}{\partial q} \\
&= \begin{cases} -\xi(q - q_{goal}), & \text{if } \| q - q_{goal} \| \leq \rho \\ -\xi\rho\frac{(q-q_{goal})}{\|q-q_{goal}\|}, & \text{if } \| q - q_{goal} \| > \rho \end{cases} \qquad (2)
\end{aligned}
$$

Repulsive forces are generated by the presence of obstacles. In the discussion of potential fields from [4], each convex C-obstacle is associated with a repulsive potential. This method of defining repulsive potentials is not applicable to us since we do not assume a model of obstacles in the workspace. In our case, the presence of obstacles is measured by the onboard sonar sensors. Therefore, we associate *each* sonar measurement with a repulsive potential of the following form [4]:

$$
U_{rep,i} = \begin{cases} \frac{1}{2}\eta(\frac{1}{d_i} - \frac{1}{d_c})^2, & \text{if } d_i \leq d_c \\ 0, & \text{if } d_i > d_c \end{cases} \qquad (3)
$$

where d_i is the measurement of sonar i (see Figure 1), d_c is a constant distance (cutoff distance), and η is a constant coefficient. The repulsive potential is inversely proportional to the distance from the robot to obstacles when the robot is in close vicinity of obstacles, and is zero when the robot is at least the cutoff distance from obstacles. The corresponding repulsive force is given by:

$$
{}^{r}F_{rep,i} = \begin{cases} -\eta(\frac{1}{d_i} - \frac{1}{d_c})\frac{n_i}{d_i}, & \text{if } d_i \leq d_c \\ 0, & \text{if } d_i > d_c \end{cases} \qquad (4)
$$

where the superscript r indicates that the repulsive force is represented in the robot-fixed coordinate system, and n_i is a unit vector in the beam direction of sonar i as shown in Figure 1. In the robot-fixed coordinate system, n_i is given by:

$$
n_i = \begin{bmatrix} \cos(i\pi/8) \\ \sin(i\pi/8) \end{bmatrix} \qquad (5)
$$

The total repulsive force expressed in the robot-fixed coordinate system is

$$
{}^{r}F_{rep} = \sum_{i=0}^{15} {}^{r}F_{rep,i} \qquad (6)
$$

The total force applied to the robot is the sum of the attractive force and repulsive force. The attractive force is expressed in the earth-fixed world coordinates while the repulsive force is in the robot-fixed coordinates. The two forces must be converted into a single coordinate system before they can be added. For ease of implementing real-time control algorithms on the Nomad 200 mobile robot, we compute the total force in the robot coordinates:

$$
{}^{r}F_{total} = {}^{r}F_{rep} + T(\theta)\ {}^{w}F_{att} \qquad (7)
$$

where the transformation $T(\theta)$ is given by

$$
T(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}
$$

The function `vm()` in the Nomad 200 mobile robot software library [18] is used to implement the potential field based control. `vm()` takes three arguments: the desired translation velocity, the desired steering velocity, and the desired turret velocity. The first component of ${}^{r}F_{total}$, multiplied by a proper gain, is passed as the first argument of `vm()`. The second component of ${}^{r}F_{total}$ is passed as the second and third arguments of `vm()`.

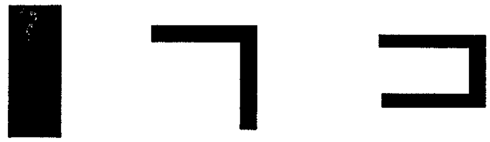

Figure 2: Three types of local-minimum-generating obstacles frequently encountered in a laboratory environment: bench, corner, and dead-end channel.

3 Modeling Laboratory Environments

The potential field based motion planning method described in the previous section is implemented on a Nomad 200 mobile robot and extensively tested in our laboratory environment. As expected, the mobile robot guided by this original potential field method is continually trapped in local minima. Based on extensive experimentation, it is observed that there are mainly three types of obstacles that mostly trap the mobile robot in local minima. These obstacles are the bench, corner, and dead-end channel as shown in Figure 2. In the case of a bench, if the goal position is on one side of the bench and the robot is on the other side, it will eventually move onto a local minimum where the repulsive force generated by the bench cancels the attractive force generated by the goal. The corner and channel types of obstacles result in local minima in a similar manner. The next section describes a method for escaping local minima created by those types of obstacles.

4 Wall-Following Method

When the mobile robot moves onto a local minimum on one side of a bench and tries to move to the other side, it is obvious that the best way to escape from the local minimum is to move around the bench.

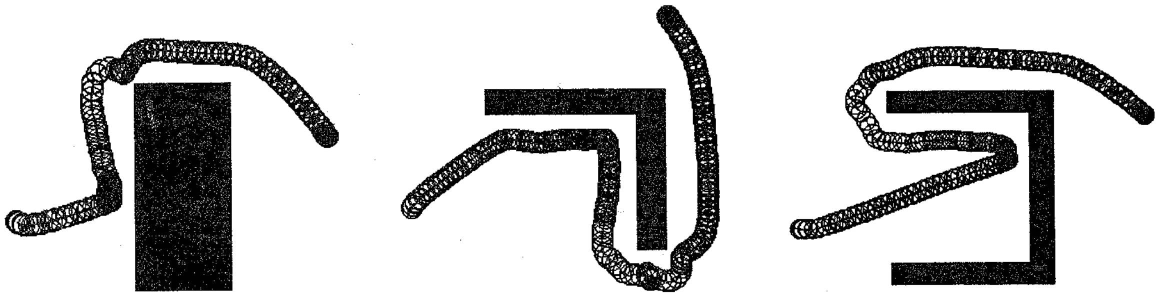

Figure 3: Escaping local minima with three types of obstacles: bench, corner, and dead-end.

For this purpose, we implement a wall-following algorithm. The algorithm takes two parameters: which-wall and distance-to-wall. The first parameter specifies that the robot follow a wall to its left or to its right. It is noted that we do not use the terminology of turning left or turning right because the robot may not necessarily directly face the bench when it stops moving at a local minimum in front of a bench. The second parameter specifies a distance that the robot should maintain from the wall to be followed.

To escape the local minimum, it is yet to be decided which way to follow the wall and when to terminate the wall-following. For off-line planning with known obstacles, it may be advantageous to follow the the shortest travel distance around the bench. Since we use the algorithm in real-time planning, in most cases the robot does not have sufficient knowledge to make any meaningful decision. Therefore, the robot arbitrarily follows a wall to its left or right. Even though the decision is arbitrary, the robot must remember its decision. As discussed later, that data will be needed in making a second attempt if the first one fails. The second decision (the switch condition) is the heart of this method. It is easy to start the wall-following control mode from a trapped local minimum, but it is less obvious as when to stop the wall-following control mode and switch back to the potential field guided control mode. We propose using the following condition. While the robot executes the wall-following algorithm, it continuously monitors the distance from its current position to the goal position. At the point where the distance starts to decrease, the wall-following control mode stops and the potential field based control mode resumes.

The switching condition described above works well for bench type of obstacles. This is because the distance increases as the robot moves away from the local minimum position while following the edge of the bench. As soon as the robot reaches the end of the bench and makes a turn, the distance starts to decrease. At this point, the potential field based control mode is able to bring the robot to its goal position. In implementation, the robot keeps moving in the wall-following mode for several iterations after the condition is met. This is to prevent the robot from falling back to the local minimum. Although the switch condition described above is strongly motivated by bench type obstacles, it also works well for two other types of obstacles, i.e., corner and dead-end channel. Figure 3 shows simulation results of the algorithm escaping local minimum in the presence of three types of obstacles. In all three plots, the initial position of the robot is at the left end of the circled robot trajectory, and the goal position is at the right end of the trajectory with a solid disc.

5 Simulation and Experimental Results

The algorithm described above was extensively tested in real laboratory environments. We present a sample of the test results and further improvements on the algorithm based on the observation of results. Figure 4 shows a simulated laboratory environment and the robot's trajectory. The experimental result with a Nomad 200 mobile robot in a similar laboratory environment is shown in Figure 5. In both simulation and experiment, the robot does not have a model of the environment. It perceives the presence of obstacles using the 16 onboard sonar sensors. The large rectangular obstacle in the middle of the workspace is a bench with a small bookcase placed at the right end. The corner-shaped obstacle at the lower part of the workspace is a wall corner. There is another bench at the upper port of the depicted workspace and a larger bookcase at the upper right corner. The starting position is at the lower end of the trajectory, or $(x, y) = (0, 0)$ in Figure 5. The goal position is at the upper end of the trajectory with a solid disc, or $(x, y) = (7.0, 14.0)$ in Figure 5.

The robot's trajectory in Figures 4 and 5 is composed of three segments. In the first segment, the robot is controlled by the potential field based algorithm. The robot moves toward the goal and reaches a local minimum in front of the bench where the attractive force and repulsive force cancel each other. If the robot velocity is low for a consecutive number of iterations and the robot position is not at the goal, the robot recognizes that it is in a local minimum. At this point, the robot switches to the wall-following control mode. In Figures 4 and 5, the robot decides

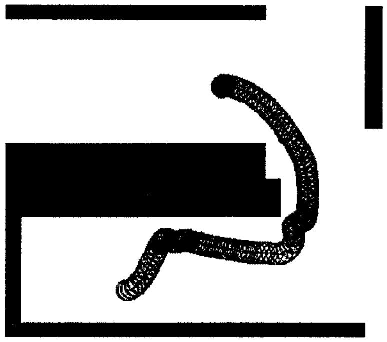

Figure 4: Simulation of the robot trajectory in a laboratory environment. After reaching the local minimum, the robot follows a wall to its left.

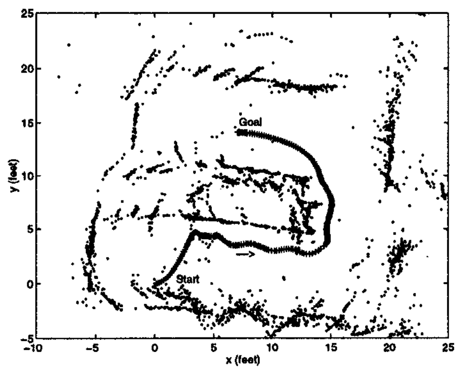

Figure 5: Experimental trajectory of the robot. After reaching the local minimum, the robot follows a wall to its left.

to follow a wall to its left. The wall-following motion constitutes the second segment of the robot trajectory. When the robot reaches the end of the bench where the small bookcase is located, the distance to the goal starts to decrease. The robot switches back to the potential field based control again. The third segment of the trajectory starts from the corner of the bench and ends at the goal position. It is observed that the simulation and experimental results are strongly consistent.

After reaching the local minimum in front of the bench, what happens if the robot decides to follow the wall to its right instead? As illustrated in Figures 6 and 7, the robot moves along the bench and makes a left turn when it encounters a wall in front of it. It makes another left turn shortly afterward. As soon as it makes the second left turn, the distance to the goal position starts to decrease and the potential field based control takes over. The robot moves toward the goal. Before long, it falls into the same local minimum again. Recognizing that it was here before, the robot decides to follow the wall in a different direction. This is why the robot needs to remember its decision as mentioned before. The remaining part of

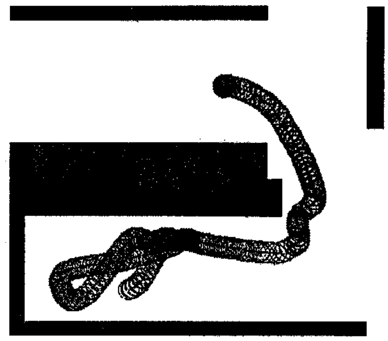

Figure 6: Simulation of the robot trajectory in a laboratory environment. After reaching the local minimum, the robot follows a wall to its right and returns to the same local minimum. It then follows a wall to its left.

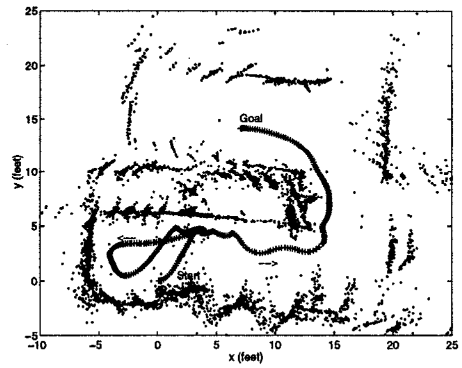

Figure 7: Experimental trajectory of the robot, first following a wall to its right and then following a wall to its left.

the trajectory is similar to that in Figures 4 and 5.

Theoretically, the local minimum encountered twice by the robot is exactly the same one. But the robot does not stop at the same location each time it falls into this local minimum. Due to cancellation, the total force is almost zero in the vicinity of local minima. Because of uncompensated friction, the robot stops moving when the total force is sufficiently small. A threshold circle is selected to determine if two local minima are in fact the same one.

Figures 4 through 7 demonstrate one of many successful experiments. The algorithm is very effective in escaping local minima encountered in laboratory environments. Nevertheless, it may fail to escape local minima in rare cases, one of which is shown in Figure 8. In this case, after trapped in the local minimum the robot makes the first attempt by following a wall to its right. As seen from the figure, the robot returns to the same local minimum. It makes the second attempt by following the wall to its left, which fails again. At this point, the algorithm reports the failure. It should be pointed out that more heuristic rules can be built into the algorithm to escape this type of local minimum,

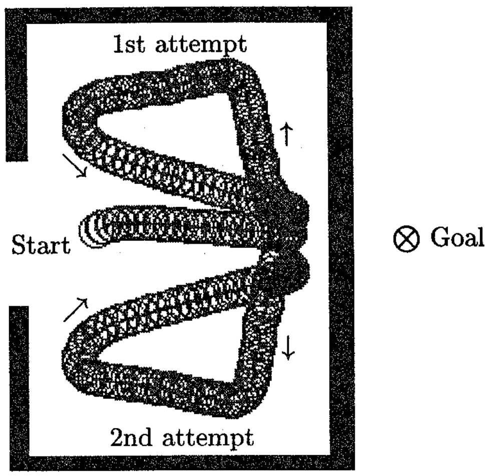

Figure 8: A scenario where the robot fails to escape from the local minimum. After reaching the local minimum, the robot first follows the wall to its right. After returning to the same local minimum, it tries to follow the wall to its left.

but increased complexities may over-shadow improvements.

It is worth mentioning an interesting experimental experience. There is a button on the top of the Nomad 200 mobile robot to turn off driving motors in case of an emergency. A person who tries to push the emergency button appears as a close-in obstacle to the robot. Instead of stopping the robot, the attempt of activating the emergency button will "scare" and consequently speed up the robot.

6 Conclusion

A simple and practical algorithm for escaping local minima in the potential field based motion planning is described and demonstrated with simulation and experimental results. The algorithm switches between two control modes: the potential field based control mode and the wall-following control mode. The main contribution of the algorithm lies in its implementation of a simple switch condition. As shown by sample trajectories from simulations and experiments, the algorithm is effective in escaping local minima in reasonably complex environments.

7 Acknowledgment

This work is supported in part by NSF grant IRI-95-96026 and CDA-95-96021. The authors would like to thank Todd West for refining and improving the wall-following program, and Art Neumann for his assistance in conducting experiments and proof-reading a draft of this paper.

References

[1] V.J. Lumelsky and A.A. Stepanov. Path-planning strateties for a point mobile automation moving amidst obstacle of arbitrary shape. *Algorithmica*, 2:403–430, 1987.

[2] D.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, Raleigh, NC, March 1987.

[3] J.F. Canny. *The Complexity of Robot Motion Planning.* MIT Press, Cambridge, MA, 1988.

[4] Jean-Claude Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991.

[5] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179–1187, Sep/Oct 1989.

[6] K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson. *Algorithmic Foundations of Robotics.* A K Peters, Wellesley, MA, 1995.

[7] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[8] B.H. Krogh. A generalized potential field approach to obstacle avoidance control. In *Proceedings of SME Conference on Robotics Research*, Bethlehem, PA, August 1984. SME Paper No. MS84-484.

[9] E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

[10] C. De Medio and G. Oriolo. Robot obstacle avoidance using vortex fields. In S. Stifter and J. Lenarcic, editors, *Advanced in Robot Kinematics*, pages 227–235. Springer-Verlag, 1991.

[11] 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, Cincinnati, OH, May 1990.

[12] R. Volpe and P. Khosla. Artificial potential with elliptical isopotential contours for obstacle avoidance. In *Proceedings of the 26th IEEE Conference on Decision and Control*, pages 180–185, Los Angeles, December 1987.

[13] J. Barraquand and J.-C. Latombe. A monte-carlo algorithm for path planning with many degrees of freedom. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1712–1717, Cincinnati, Ohio, May 1990.

[14] F. Lamiraux and J.P. Laumond. On the expected complexity of random path planning. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 3014–3019, Minneapolis, Minnesota, April 1996.

[15] Yilin Zhao. Theoretical and experimental evaluation of local minium recovery navigation algorithm. In Yuan F. Zheng, editor, *Recent Trends in Mobile Robots*, chapter 4, pages 75–117. World Scientific, 1993.

[16] I. Kamon, E. Rivlin, and E. Rimon. A new range-sensor based globally convergent navigation algorithm for mobile robots. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 429–435, Minneapolis, MN, April 1996.

[17] Hsuan Chang. A new technique to handle local minimum for imperfect potential field based motion planning. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 108–112, Minneapolis, Minnesota, April 1996.

[18] Nomadic Technologies, Inc. Language reference mannual. Technical report, 2133 Leghorn Street, Mountain View, CA 94043, 1995.