Presented at the ISRAM Conference, Montpellier, France, May 27-30, 1996.

OmniNav: Obstacle Avoidance for Large, Non-circular, Omnidirectional Mobile Robots

B. Holt[1], J. Borenstein[1], Y. Koren[1], and D. Wehe[2]

The University of Michigan

1) Department of Mechanical Engineering and Applied Mechanics

2) Department of Nuclear Engineering and Radiological Science

Ann Arbor, MI 48109-2110, USA

ABSTRACT

This paper describes an obstacle avoidance method designed specifically for large, non-circular robots with omnidirectional motion capabilities. The method, called OmniNav, is reflexive, meaning that it is able to make navigation decisions without having to discern details of the environment, such as edges. The OmniNav method combines elements of the previously developed Virtual Force Field (VFF) and Vector Field Histogram (VFH) methods to protect arbitrarily shaped mobile platforms from collisions. To this end the method computes a preliminary direction of travel based on the VFH method, which assumes the robot was point-sized. Then, highly localized potential fields are applied to "act-on points" around the robot's real perimeter. A transfer function uses these "virtual forces" to alter the preliminary direction of motion and to add rotation as necessary to avoid collisions.

Keywords: Mobile robots, Obstacle avoidance, Omnidirectional, Ultrasonic sensors.

1. INTRODUCTION

The specific problem addressed by our system is how to guide an irregularly shaped three-degree-of freedom vehicle around obstacles. Our method, called OmniNav, has three unique features:

- OmniNav works in uncertain environments. Previously, an effective way of dealing with large-robot navigation has been to represent obstacles in the workspace of the robot as configurations of the robot that are not allowed. This approach, called the configuration space approach, was first described by Lozano-Perez [1987]. In this approach, it is necessary to accurately locate the edges of all obstacles in the workspace, and then either change their location or search for robot configurations which intersect these edges. In contrast, the OmniNav method uses a histogramic map representation of the environment (see Section 2), which does not require knowledge of specific obstacle features.

- OmniNav is strictly reflexive and can operate independently of any higher-level navigation architecture. By using global path planning, many researchers have developed methods that optimize the orientation of an irregular vehicle with respect to the obstacles in the environment. This is usually done by constructing a Voronoi Diagram and applying potential fields to multiple act-on points on the perimeter of the robot [Hague et al., 1989; Barraquand et al., 1992]. All of these systems operate in an open loop, i.e., they compute the path once prior to moving and not again. Our method, on the other hand, operates in a closed loop, updating the control commands at a rate of approximately 25 Hz. Thus, OmniNav is able to continually sample its environment and upgrade its map.

- OmniNav controls three degrees of freedom. As discussed by Borenstein and Raschke [1992], a similar method has been successfully applied to two degree-of-freedom vehicles, such as the University of Michigan's CARMEL. OmniNav is able to control not only the translation of the vehicle, but also its rotation, independently. The additional degree-of-freedom allows it to pass through areas that would not be traversable with only two degrees-of-freedom.

The following two sections discuss in detail the OmniNav method. Section 2 explains how the VFF method is applied to act-on points around the vehicle. In Section 3 the VFH method, used to initially find open pathways, is explained. Section 4 discusses the transfer function that adjusts the direction of travel as selected by VFH and optimizes its orientation in response to the VFF forces. Section 5 gives the results in simulation for OmniNav and describes our current effort to employ OmniNav on a real mobile robot.

2. THE VIRTUAL FORCE FIELD (VFF) METHOD

The VFF method is specifically designed to accommodate and compensate for inaccurate range readings from ultrasonic or other sensors [Borenstein and Koren, 1989]. To do so, it uses a two-dimensional Cartesian grid, called the histogram grid C, to represent data from ultrasonic (or other) range sensors. Each cell (i,j) in the histogram grid holds a certainty value (CV) $c_{i,j}$ that represents the confidence of the algorithm in the existence of an obstacle at that location. This representation was derived from the certainty grid concept that was originally developed by Moravec and Elfes, [1985]. In the histogram grid, CVs are incremented when the range reading from an ultrasonic sensor indicates the presence of an object at that cell.

Combining the histogram grid with the potential field concept, the VFF method allows the immediate use of real-time sensor information to generate repulsive force fields. Fig. 1 illustrates this approach: As the vehicle moves, a square "window" accompanies it, overlying a region of C. We call this region the "active region" (denoted as C*), and cells that momentarily belong to the active region are called "active cells" (denoted as $c^*_{i,j}$). The window is always centered about the robot's position.

Each active cell exerts a virtual repulsive force $F_{i,j}$ toward the robot. The magnitude of this force is proportional to $c^*_{i,j}$ and inversely proportional to d^n, where d is the distance between the cell and the center of the vehicle, and n is a positive number (usually, n = 2). All virtual repulsive forces add up to yield the resultant repulsive force F_r. Simultaneously, a virtual attractive force F_t of constant magnitude is applied to the vehicle, "pulling" it toward the target. Summation of F_r and F_t yields the resultant force vector R. The direction of R is used as the reference for the robot's steering command.

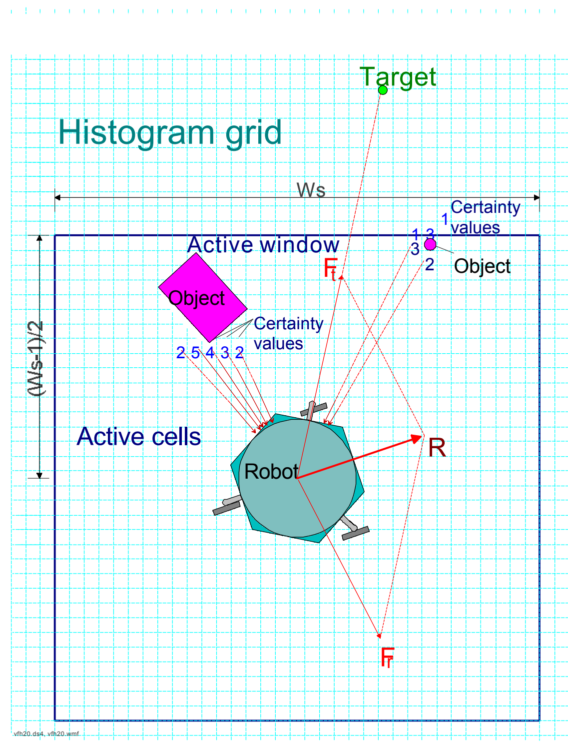

Figure 1: The Virtual Force Field (VFF) concept: Occupied cells exert repulsive forces onto the robot, while the target applies an attractive force.

In the course of earlier experimental work with the VFF algorithm it was found that a robot guided by this method alone tended to develop oscillations when in cluttered environments or traveling in down narrow passages. Koren and Borenstein, [1991] introduced a mathematical analysis of these inherent problems. As this analysis shows, one main contributing factor to instability of motion with potential field methods (PFMs) is the non-linear force function of the repulsive forces. However, non-linearity is employed by most PFMs to assure that the repulsive force is strong enough to be effective at a certain distance from the obstacle, so that the robot can begin an avoidance maneuver in time.

3. THE VECTOR FIELD HISTOGRAM (VFH)

To overcome these problems, Borenstein and Koren [1991] developed a new obstacle avoidance method called the vector field histogram (VFH). The VFH method builds the histogram grid the same way the VFF method does. However, the VFH method then introduces an intermediate data representation called the polar histogram. The polar histogram retains the statistical information of the histogram grid (to compensate for the inaccuracies of the ultrasonic sensors), but reduces the amount of data that needs to be handled in real-time. This way, the VFH algorithm produces a sufficiently detailed spatial representation of the robot's environment for travel among densely cluttered obstacles, without compromising the system's real-time performance.

The polar histogram H is an array comprising 72 elements; each element represents a 5°-sector of the robot's surroundings. During each sampling interval, the active region of the histogram grid C* is mapped onto H as shown in Fig. 2, resulting in 72 values that can be interpreted as the instantaneous polar obstacle density around the robot.

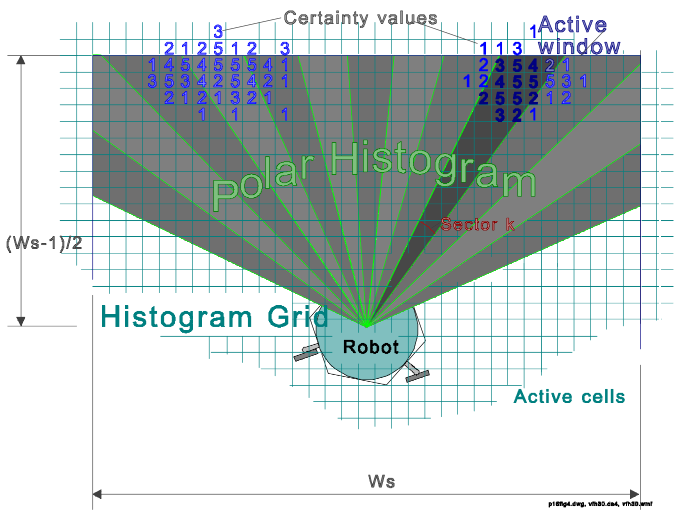

Figure 2:
Mapping the histogram grid onto the polar histogram

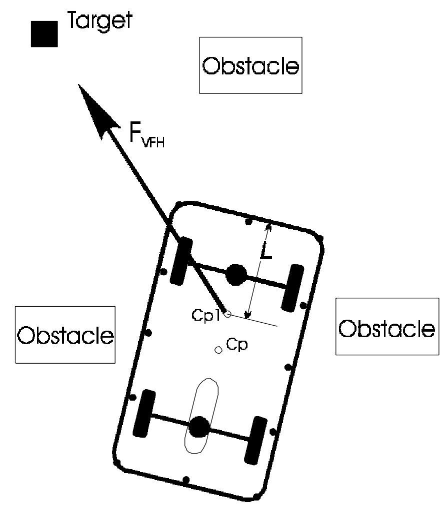

Figure 3:
VFH selects a direction of travel

After the polar histogram has been constructed, the VFH algorithm computes the required steering direction for the robot, Θ. A polar histogram typically has peaks (sectors with high obstacle density), and valleys (sectors with low obstacle density). Any valley with obstacle densities below a selected threshold level is a candidate for travel. Since there are usually several candidate-valleys, the algorithm selects the one that most closely matches the direction to the target. Note that the width of a valley, W_V, can be measured in terms of the number of consecutive sectors that are below threshold. A small W_V indicates a narrow passageway or corridor. For narrow valleys, i.e., closely spaced obstacles, Θ is chosen to be the center of the valley.

4. OMNINAV FOR NON-CIRCULAR, OMNIDIRECTIONAL ROBOTS

While both VFF and VFH methods were originally designed for point-sized robots, a combination of these methods can be used to efficiently guide non-point mobile robots through densely cluttered obstacle courses. The combination of the two methods allows us to use each one to its advantage, while avoiding the disadvantages. For example, since stable motion and better spatial resolution are the strength of the VFH method, it is used to determine what we call the "principal steering direction." The VFF algorithm, on the other hand, can provide local corrective measures to account for the shape of the vehicle. Limiting the potential fields-based VFF method to a corrective function, we can use a steep force profile with only short-range effects. This way we reduce the oscillatory tendency of PFM-based obstacle avoidance. The following discussion explains in more detail how the OmniNav system works.

a) Computing the principle steering direction **Θ** with the VFH method (Figure 3)

The task of the VFH component is to find the openings through which the robot should travel to reach the goal. To do so, the VFH algorithm is applied at a point CP_1 to determine the principal steering direction Θ. A vector F_{VFH} is computed and applied in that direction. CP_1 is located on the longitudinal axis of the vehicle, but its optimal location (in terms of distance L from the front of the vehicle) differs for different vehicles.

b) Local protection with the VFF method (Figure 4)

The VFF method is applied to each one of the n act-on points A_n on the periphery of the robot. Technically, this is done for each act-on point A_n by adding up all individual repulsive forces $F_{i,j}$ from filled cells in the histogram grid, yielding n repulsive forces F, (note that Fig. 4 does not show the histogram grid explicitly). The force fields that act on each act-on point are very steep; in our application they are generated by $F_{i,j} \propto 1/d^5$. This way, the effective range is very short. Consequently, $F_k \neq 0$ only when an act-on point is very close to an obstacle. To avoid having the repulsive forces impact the speed of the vehicle, the force component that is parallel to the direction of travel is eliminated.

c) Extracting the relevant information

As can be seen in Figure 5, there are times when two adjacent act-on points straddle an obstacle and may thus generate conflicting signals. In the first part of Figure 5 the robot is traveling directly to the right, approaching the small obstacle. After the repuslive forces are projected perpendicular to the direction of travel, the two remaining force vectors tend to cancel each other. To reduce this problem, the repulsive forces felt by each act-on point are always oriented toward the center of the robot. This way, the repulsive forces protect the whole robot, instead of each act-on point individually.

In calculating the moments on the robot, however, a different treatment is needed. At the corners, the moments are found by computing the cross product of the VFF-generated force with the arm from the center point of the robot to the act-on point. In the third illustration of Figure 5 it can be seen that forces along the side of the vehicle, if computed in the same manner, will actually rotate it toward an obstacle. Thus, at the side act-on points, the components of force along the length of the robot are eliminated from the moment calculation.

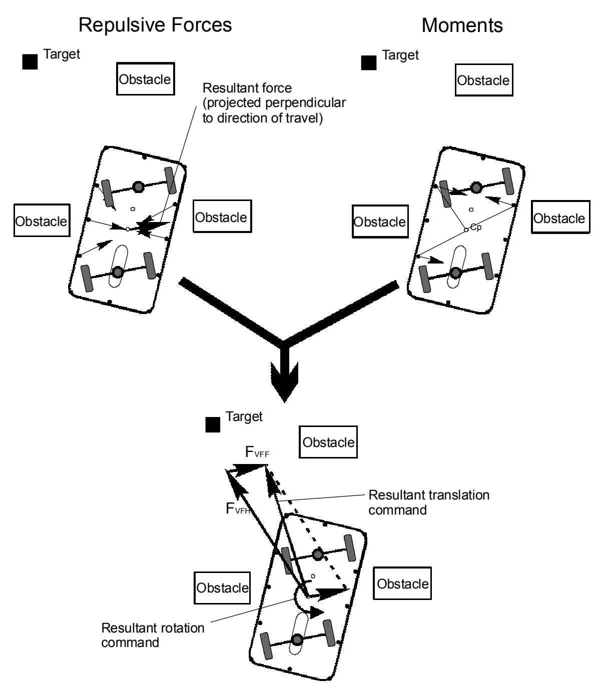

Figure 4: Finding forces and moments from VFF applied to the act-on points. Summation with the VFH-based direction yields the final direction of travel.

d) Manipulating the correction information

The kinematic control program for the robot and the simulation is designed to accept translation velocity and orientation commands [Borenstein, 1995]. The translation command begins with the summation $V = aF_{VFH} + bF_{VFF}$, where F_{VFH} is the vector found by the VFH method, and F_{VFF} is the sum of the repulsion forces found by the VFF method (see Fig. 6). The coefficient a is determined as $a = 1/W_V$, i.e., the inverse of the width of the candidate-valley selected by the VFH method (see Sect. 3.2). The effect of this latter scaling factor is as follows: when the robot circumnavigates a single obstacle, the candidate-valley will be wide and a will be small. Consequently, the correctional effect of F_{VFF} will be quite significant. In a narrow corridor, on the other hand, the robot is forced to be close to the walls and very large repulsive forces develop (because of $F_{i,j} \propto 1/n^5$). Furthermore, small diversions of the robot from the centerline result in dramatic fluctuations of the forces and consequently in oscillatory motion. However, a will be relatively large and will dominate the resultant vector F_S. This way, the oscillatory behavior usually associated with potential field control is avoided.

The orientation command begins with a sum of moments $W = aW_{VFH} + bW_{VFF}$. W_{VFF} is completed from the forces on each act-on point, and W_{VFH} is the angular difference between the vehicle's current orientation and the VFH. The coefficient b indicates the tendency of the robot to align itself in the free direction. This helps avoid any situation where the robot straddles an opening and generates conflicting moments from the act-on points to either side (see Fig. 5).

After summing the moments and forces from VFH and VFF, the two commands are then normalized with respect to each other as follows:

$$|V| = \frac{d \bullet V}{\sqrt{V^2 + W^2}}$$

$$|W| = \frac{e \bullet W}{\sqrt{V^2 + W^2}}$$

Where e and d are weighting coefficients.

The effect of this normalization is to slow the robot's translation down when experiencing large moments, giving it time to adjust its orientation as necessary before proceeding. It also provides a means of keeping the commands to levels that are controllable by the kinematic control algorithm [Borenstein 1995].

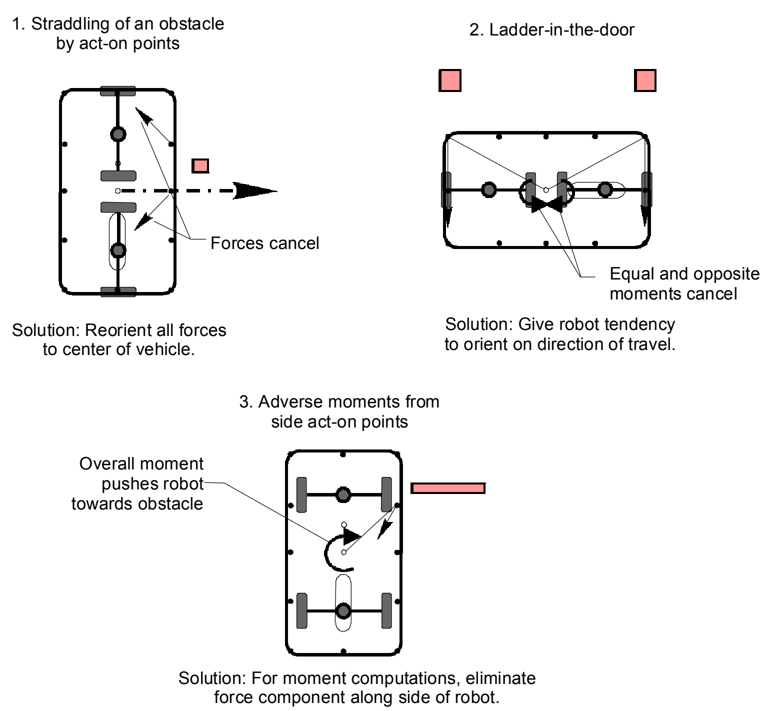

Figure 5: Special cases in the application of the OmniNav method and their heuristic solutions.

5. SIMULATION RESULTS AND FUTURE TESTBED

To verify the performance of the OmniNav method, a multi-degree-of-freedom robot, called the Compliant Linkage Autonomous Platform with Position Error Recovery (CLAPPER), will be employed as the platform. The CLAPPER is a four-degree-of-freedom robot, consisting of two differential-drive, TRC LabMate robots in tandem, connected by a compliant linkage. The controller of the CLAPPER coordinates the motion of the two TRC trucks and assures smooth, omnidirectional motion [Borenstein 1995]. Figure 6 shows a photograph of the CLAPPER, which is currently being instrumented with 32 ultrasonic sensors. The results presented in this section, however, are simulation results only. We expect to have actual performance results from the real robot available for presentation at the conference.

5.1 Simulation Setup

The OmniNav method was simulated by adapting the simulator used to develop the CLAPPER. In this simulation, an obstacle course can be created and is then converted into a histogramic map, from which the VFH and VFF commands can be generated. The simulated robot uses odometry and OmniNav obstacle avoidance to move from an arbitrary starting point to an arbitrary target. Figure 7 shows the type of virtual obstacle used to simulate the program. The square poles are 100mm on a side, and are spaced such the most optimal paths among them would have at most 25 cm of combined clearance on both sides of the robot. This minimum spacing is consistent with the ultrasonic sensor array to be used on the real robot, in which the transducers have a minimum range of 27 cm.

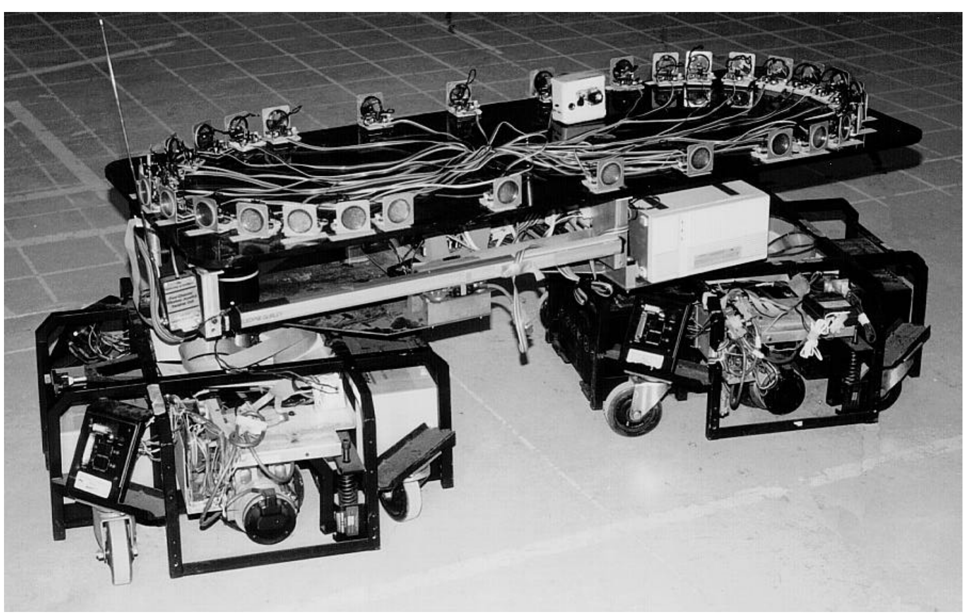

Figure 6: The University of Michigan's CLAPPER platform is currently being equipped with 32 ultrasonic sensors for testing the OmniNav method.

5.2 Simulation Results

A sample run through the previously described obstacle course is shown in Figure 7. With the OmniNav obstacle avoidance algorithm the simulated robot did maintain a minimal clearance of 10 cm from all obstacles at all times. On a 486/50 MHz PC, the average sample time was less than 20 ms, and the simulated velocity of the robot averaged 230 mm/s.

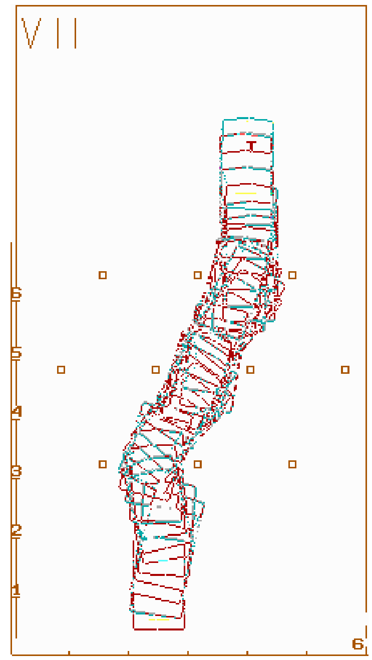

Figure 7: Simulation results with the OmniNav method.

6. CONCLUSIONS

In this paper we have introduced the OmniNav reflexive obstacle avoidance method. The OmniNav method combines the previously developed VFH and VFF methods and can be applied to robots of arbitrary shape to guide them through densely cluttered obstacle courses. The system has been verified by simulation and is currently being installed on a real test platform.

Acknowledgments

This research was funded by the Department of Energy grant DE-FG02-86NE37969. The authors wish to thank Brian Costanza, Liquang Feng, Chris Minekime, and Iwan Ulrich for their many useful suggestions.

7. REFERENCES

Barraquand, J.,Langloi, B., and Latombe, J.C., 1992, “Numerical Potential Field Techniques for Robot Path Planning.” IEEE Transactions on Systems, Man, and Cybernetics, Vol. 22, No. 2, pp. 224-241.

Borenstein, J. and Koren, Y., 1989, “Real-time Obstacle Avoidance for Fast Mobile Robots.” IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, Sept/Oct, pp. 1179-1187.

Borenstein, J, and Koren, Y., 1990, “Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments.” Proceedings of the IEEE Conference on Robotics and Automation, Cincinnati, Ohio, May 13-18, pp. 572-577.

Borenstein, J. and Koren, Y., 1991, “Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation.” Proceedings of the IEEE Conference on Robotics and Automation, Sacramento, California, April 7-12, pp. 1398-1404.

Borenstein, J. and Koren, Y., 1991, “The Vector Field Histogram—Fast Obstacle-Avoidance for Mobile Robots.” IEEE Journal of Robotics and Automation, Vol.7, No.3, pp. 278-288.

Borenstein, J. and Raschke, U., 1992 “Real-time Obstacle Avoidance for Non-Point Mobile Robots.” The Fourth World Conference on Robotics Research.

Borenstein, J. 1995, “Control and Kinematic Design of Multi-Degree-of-Freedom Mobile

Robots with Compliant Linkage” IEEE Transactions on Robotics and Automation, Vol. 11, No. 1, February, pp. 21-35.

Hague, T., Brady, M., and Cameron, S., 1990, “Using Moments to Plan Paths for the Oxford AGV.” Proceedings of the 1990 IEEE International Conference on Robotics and Automation, Cincinnati, Ohio, May 13-18, pp. 210-215.

Lozano-Perez, T., 1987, “A Simple Motion-Planning Algorithm for General Robot Manipulators.@ IEEE Journal of Robotics and Automation, Vol. RA-3, No. 3, pp. 224-238.

Moravec, H.P. and Elfes, A., 1985, “High Resolution Maps from Wide Angle Sonar.” IEEE Conference on Robotics and Automation, Wahington D.C., pp. 116-121.