Proceedings of the 30th Conference
on Decision and Control
Brighton, England • December 1991

MICROPROCESSOR-BASED CONTROL OF AUTONOMOUS MOBILE ROBOTS WITH OBSTACLE AVOIDANCE

Han-Pang Huang and Pei-Chien Lee
Robotics Laboratory
Department of Mechanical Engineering
National Taiwan University
Taipei, Taiwan 10764, R.O.C.

Abstract

A real-time obstacle avoidance algorithm for autonomous mobile robots is first developed. The algorithm is sensor-based and consists of H-mode and T-mode. Then the algorithm is implemented on the NTU II mobile robot, which is fabricated by ourselves and microprocessor-based control. It will be shown that the algorithm is more efficient and more robust than other sensor-based algorithms. In addition, the algorithm will guarantee a solution for the obstacle avoidance problem. Since the algorithm only takes up a small computational time, it can be implemented in real time.

I. Introduction

The obstacle avoidance algorithm in this paper is aimed at mobile robots that move in a stationary environment. Such problem can be regarded as a two-dimension obstacle avoidance problem. In general, the obstacle avoidance algorithms can be divided into two categories: model-based [2,3,9-14,17-19] and sensor-based [1,4-8,15,16]. The model-based algorithm uses a built-in obstacle model to generate obstacle-free path, while the sensor-based algorithm utilizes sensors to gather environment informations for obstacle avoidance. Generally speaking, a model-based system considers the problem in a global way. All obstacles in the working environment can be completely described. Hence, an optimal obstacle-free path can be found by the model-based algorithm. However, a complete and precise model for obstacles is very difficult to establish. Due to the features of the model and computational complexity, the model-base algorithm is not suitable for time varying environment and practical use. On the other hand, a sensor-based system usually makes decision locally; thus, it cannot ensure to find a solution for the problem. Since the real world is complex and may be time-varying, only sensors can provide an efficient and precise information of obstacle arrangement. However, the solution to the obstacle avoidance problem should be justified in sensor-based systems.

In this paper, a real-time, sensor-based algorithm is developed for collision avoidance of mobile robots. The algorithm is composed of a heuristic mode (H-mode) and a track mode (T-mode). The existence of an obstacle-free path by using this algorithm will be justified. Then, the algorithm together with ultrasonic sensors is implemented on our NTU II mobile robot. The knowledge on sensory feedback will be discussed. Finally, computer simulation and implementation results are illustrated, and conclusion is followed.

II. Obstacle Avoidance Algorithm

In the problem discussed, any obstacle has finite area and finite boundary length. Obstacles are time-invariant. No information on the geometry or positions of obstacles are available. The robot is considered as a point in a two dimensional plane. The central theme of our approach is the sensor-based algorithm. The lack of information about obstacles is compensated by the sensor feedback. Hence, the robot is able to detect the fact of meeting obstacles and either to bypass or to track along the obstacle boundary. In short, the algorithm will consist of both H-mode and T-mode.

Heuristic Mode

First, assume that there are only convex obstacles scattered in the environment. An obstacle is convex if the straight line between any two points on the obstacle's boundary still lies within the obstacle. The scattering of obstacles is given in Fig.1; single obstacle and multiple obstacles are shown there. The mobile robot traverses from the starting point S to the target T. During the maneuvering, the robot may encounter obstacles. No matter how obstacles are scattered, the two flanks of an obstacle can be found. If the flanks of an obstacle are hidden by other obstacles (e.g. Ob_1 in Fig.1b), then the obstacle is ignored. Otherwise, a heuristic search is employed for finding the moving direction [4,11]. For instance, a heuristic function is applied to directions $\overrightarrow{L_1}, \overrightarrow{L_2}, \overrightarrow{R_1}, \overrightarrow{R_2}$, then the best one is selected as the forward direction for the robot.

In general cases, there are many obstacles and some of them are concave. The H-mode cannot work well. The robot may be trapped in some regions. The reason is that the target T must be located at global extrema in the heuristic function approach. However, the heuristic function may have several local extrema due to the distribution of the obstacles and the limitation of the ranging ability of sensors. When the robot is maneuvered to one of the local extremum, the robot will be "boxed" in that region. Thus, the H-mode cannot bring the robot to the target. So we need a non-heuristic method to solve the remaining problems.

Track Mode

The track mode is similar to the Bug 2 algorithm proposed by Lumelsky [15]. However, our method is more efficient and results in shorter traveling distance than Lumelsky's method. The justification can be found in our previous paper [8]. In addition, the concept of hit points and leave points will be used. The hit point H_i on an obstacle boundary is the point that the robot meets the obstacle while maneuvering toward the target. After the hit point, the robot must move along the boundary of the obstacle while keeping a safe distance. Though there are two directions at the hit point for the robot to track along the boundary, either direction can be chosen. Since the robot need not encircle the obstacle forever, some decision conditions should be imposed to stop tracking and leave the obstacle's boundary. The point that the robot leaves the obstacle boundary and starts moving in free space is called the leave point. The leave point L_i will exist if L_i, H_i, and the target point T lie on a straight line, where L_i is between H_i and T. Or equivalently,

$$\angle H_i L_i T = 180° \tag{1}$$

After reaching L_i, the robot moves along a straight line to the target. The hit points H_i and leave points L_i must be found in pairs or the problem has no solution. If H_i exists but cannot find L_i, then the robot will maneuver back to H_i again. This implies that either the target or the robot is enclosed by the obstacle. In such case, no obstacle-free paths exist.

Eq.(1) guarantees that the robot will approach T after passing a pair of hit point and leave point. Notice that S, T, the hit points and leave points all lie on $\overline{ST}$. By using the rule, we can prove that the total path length is finite no matter how complex the environment

is. This implies that the obstacle-free path can be found. It is shown below.

Let D be the total path length; OB_n be the n-th obstacle met by the robot; and B_n be the perimeter of Ob_n. Assume that there is only one obstacle met by the robot. Despite the shape of the obstacle, the boundary length connecting a pair of H_i and L_i is less than the perimeter of the obstacle. In such case, the upper bound of the total path length can be obtained as

$$D \leq |\overline{ST}| + B_1 \tag{2}$$

where $|\overline{ST}|$ is the length between the starting and the target locations. If the number of hit points is k_1, then we can infer that

$$D \leq |\overline{ST}| + k_1 B_1 \tag{3}$$

Next, consider a complicated and concave obstacle, as shown in Fig.2. In this case the robot will hit the obstacle twice at H_1 and H_2, respectively. The corresponding leave points are L_1 and L_2. Since the forward direction at each hit point is not specified, there are four possible solutions. The most efficient one is shown in Fig.2a. The worst one is shown in Fig.2b. Since the number of hit points is two, each segment on the obstacle boundary will never be passed more than twice. Hence, the total path length is bounded by Eq.(3).

In general cases, the robot may meet many obstacles and hit each obstacle many times. Let k_n be the number of times that the robot hits the obstacle Ob_n, and K be the total number of hit points. Thus, $K = \sum k_n$. It can be inferred from Eq.(3) that the total path length is bounded by

$$D \leq |\overline{ST}| + \sum_{n=1}^{N} k_n B_n \tag{4}$$

where N denotes the number of obstacles met by the robot.

The total path length D will be bounded if both k_n and B_n are bounded. Since we assume that the dimension of the obstacles is bounded, B_n is bounded. By the definition of H_i and L_i, we get

$$\begin{aligned} &|\overline{ST}| \geq |\overline{H_1T}| > |\overline{L_1T}| \geq |\overline{H_2T}| > |\overline{L_2T}| \\ &\geq \cdots \geq |\overline{H_KT}| > |\overline{L_KT}| \geq 0 \end{aligned} \tag{5}$$

Thus,

$$|\overline{ST}| > \sum_{i=1}^{K} (|\overline{H_iT}| - |\overline{L_iT}|) \tag{6}$$

where $|\overline{H_iT}| - |\overline{L_iT}|$ can be regarded as the thickness of the obstacle. This value cannot be infinitely small or it will be physically meaningless. Let $|\overline{H_iT}| - |\overline{L_iT}| \geq \epsilon > 0$ for all i. Then

$$|\overline{ST}| > K\epsilon \tag{7}$$

Since $|\overline{ST}|$ and ϵ are bounded, K is bounded. Each k_n and N are also bounded. This proves that total path length D in Eq.(4) is a finite value. Hence, the obstacle-free path can be found. However, the upper bound defined in Eq.(4) is very loose. In most cases, the total path length will be much smaller than this upper bound.

Notice that the proof is based on the assumption: the obstacles are finite dimension but not infinitely thin. Eq.(1) guarantees the convergence in Eq.(5). However, the relation between K and the properties of obstacles, e.g. shape and distribution of obstacles, is not considered. It can be further shown that the upper bound given by Eq.(4) is tighter than that given by Lumelsky [8].

Combination of H-Mode and T-mode

The H-mode is good for convex obstacles or small concave obstacles. In particular, if obstacles are small relative to the effective range of the sensors, the H-mode works well. If there are large and/or concave obstacles, the robot may be boxed-in and wanders in some region. In such case, the T-mode should take over. Thus, the switching condition from the H-mode to the T-mode should be set up. Though the T-mode guarantees a solution, it is not the case for the H-mode. The solution existence of the combination of the H-mode and T-mode should be justified.

When the robot moves from S or departs from L_i toward the target, the H-mode is applied rather than moves along a straight line. At each time, a sequence of median points $Q_0, Q_1, Q_2, \cdots, Q_j, Q_{j+1}, \cdots$ are generated by the H-mode, where Q_0 may be the starting point S or any leave point. The robot moves along a straight line between two consecutive median points Q_j, Q_{j+1}. It will be shown that an obstacle-free path can be found by using the algorithm if the H-mode satisfies the condition in the following theorem. The condition that switches from the H-mode to the T-mode is developed as well.

Theorem 1:
In H-mode, if the next median point Q_{j+1} and the current location Q_j of the robot satisfy the conditions

$$|\overline{Q_jT}| - |\overline{Q_{j+1}T}| > \delta > 0, \tag{8}$$

and the robot moves along a straight line between Q_j and Q_{j+1}, then the combination of H-mode and T-mode still guarantees that the total path length is finite. Here δ is a significant value and cannot be infinitely small.

proof:

If Q_j, $j = 1, \cdots, p$, can always be found and satisfy Eq.(8), then

$$|\overline{Q_0T}| = |\overline{Q_pT}| + \sum_{j=1}^{p} (|\overline{Q_{j-1}T}| - |\overline{Q_jT}|) > |\overline{Q_pT}| + p\delta. \tag{9}$$

where p is the number of median points. Hence,

$$p < \frac{|\overline{Q_0T}| - |\overline{Q_pT}|}{\delta} \tag{10}$$

Because δ cannot be infinitely small, p is finite. The path generated by H-mode is proportional to p. Namely, the path length generated by the H-mode is finite. In addition, if p is large enough such that

$$|\overline{Q_pT}| < |\overline{Q_0T}| - p\delta < d \tag{11}$$

where d is the advancing distance in each step, then the median point Q_{p+1} is chosen to be T. The sequence of median points Q_j converges to T and the target is arrived.

If Q_{j+1} cannot satisfy Eq.(8), then Q_j is defined as a hit point and the tracking procedure (T-mode) is applied. T-mode is applied until Eq.(1) is satisfied; then a leave point is defined and H-mode governs the robot again. The upper bound of the total path length in Eq.(4) can be modified as

$$D \leq f_H + \sum_{n=1}^{N} k_n B_n. \tag{12}$$

where f_H is the total path length generated by the H-mode. The second term of the right hand side in inequality (12) is the contribution of T-mode and the first term is of H-mode. Because the path segments generated by H-mode are of finite length, f_H is finite. In addition, the conditions in Eq.(8) dose not affect the convergence in Eq.(5). This implies that k_n and N are still finite. Thus, if H-mode and T-mode are combined, the total path length is bounded by Eq.(12).

Compare Eq.(12) with Eq.(4). Since the H-mode can avoid simple obstacles and does better than the pure T-mode, see Fig.3, the upper bound given in Eq.(12) will be smaller than that given in Eq.(4) in practice.

The algorithm is summarized as follows.

1) Initialization, i=1, current position is the starting point S.
2) H-mode: From current position move toward target T by a heuristic method. A series of median points will be generated until one of the followings occurs.
   a) Target is reached. Stop.
   b) If the next median point generated by the heuristic method does not satisfy Eq.(8), then the current point is defined as a hit point H_i. Go to Step 3.
3) T-mode: Take a tracking rules to track along the obstacle boundary until one of the followings occurs.
   a) Target is reached. Stop.
   b) Eq.(1) is satisfied. Define a leave point L_i. Set i=i+1. Go to Step 2.
   c) The robot returns to H_i. No obstacle-free path exists. Stop.

In practice, Eq.(1) used in Step 3b is not so strict. Certain tolerance can be allowed. In Step 2a and Step 3a, the target T is reachable if T is visible, i.e., T lies in the effective range of sensors and there are no obstacles between the robot and T. In summary, the solution of obstacle avoidance consists of a heuristic approach and a tracking approach. The robot can avoid obstacles efficiently in a simple workspace by the heuristic approach. In general environment, both heuristic and tracking procedures should be used to find an obstacle-free path automatically.

Up to now, the selection of heuristic function has not been discussed. In fact, the heuristic function in H-mode can be arbitrary functions as long as the global minimum is located on the target. For example, the potential field function is one type of heuristic functions. However, any heuristic function suffers from the existence of local minima. In order to find a global solution, the obstacle boundary tracking method must take over whenever the heuristic approach fails. The tracking procedure in T-mode can be replaced by one of the tracking algorithms given in [18] as long as the algorithm guarantees the convergence of the path length. Theorem 1 gives the sufficient condition which the H-mode is switched to the T-mode, i.e., the hit point, for finding a global solution. The definition of the hit points and leaves points mentioned above can guarantee the global solution for avoiding obstacles in an unknown environment. Note that the algorithms given in [18] define the conditions which the T-mode is switched back to the H-mode.

III. Implementation of The Mobile Robot System

The above obstacle avoidance algorithm is implemented on our own NTU II mobile robot. The mobile robot is a cylindrical type with radius of 32 cm. It is 50 cm in height. The robot is driven by two stepping motors. Using the velocity difference of the two motors, the position and orientation of the robot can be controlled. Two additional idle wheels are used for supporting purpose. The outlook of the mobile robot is shown in Fig.9.

In order to obtain the local environment information at once, twelve sensors should be located at the center of the robot and arranged in radial direction. Namely, every thirty degrees will install one sensor. But, for avoiding collision, detections only on the right side, the left side and the front of the robot would be enough. Hence, only seven directions are detected instead of twelve. Furthermore, due to limited budget, only one ultrasonic sensor is installed on the robot. The sensor is mounted on a seat plate which is driven by a stepping motor. Since seven directions need to be detected, the ultrasonic sensor is rotated and stopped for one second at 22.5 degrees interval. A complete scanning cycle takes about twenty seconds. The ultrasonic sensor is OMRON E4M model. Its effective range is from 10cm to 3.5m. Since the sensor output is an analog current, the signal should be converted to digital signal by a ADC (analog to digital converter). The resolution of the range data is limited by the 8-bit ADC; i.e., about 1.5 cm.

Since the shape of the robot is symmetrical to its center line, the swept volume of the robot is unchanged as it rotates. Hence, with slight modification of the sensory information, the volume of the robot can be shrunk to a point. Let the range data measured by the sensor is r; the length of the sensor is R_s; and the robot's radius is R_b (see Fig.4a). Then, the modified range data R_m is given by

$$R_m = r + R_s - R_b \tag{13}$$

Using R_m to generate the path, we have, in fact, expanded the obstacle just like some model-based systems [14] (see Fig.4b), and shrunk the robot to a point.

A PC-XT microprocessor board is used as the controller of the robot system. There are no communication devices between the microprocessor board and other computer devices so that the robot works independently. An interface card is designed to handle the motion of the robot and sensory feedback data. The block diagram of the robot system is shown in Fig.5.

IV. Knowledge on Sensory Feedback

In the obstacle avoidance algorithm, the robot is maneuvered forward step by step. Each step, the robot advances a fixed distance. After walking a step, the robot detects the existence of obstacles by range sensors and determines the next forward direction. The mobile robot continuously generates the path based on its current coordinates and incoming local information. This procedure is repeated until the robot arrives to the target.

In both the heuristic approach and tracking approach, the sensor feedback informations from seven directions are used to generate the next median goal. Let the seven directional vectors and range data be $\overrightarrow{R_n}$ and r_n, respectively; where $n = 1 \sim 7$. If r_n is less than the effective range, r_v, of the sensors, then there are obstacles in $\overrightarrow{R_n}$ direction. Note that the effective range r_v can be reset by the control program and it is less than the physically effective range, i.e., 3.5 meters.

Consider the H-mode. First, all sensors are rotated such that one of the sensors is pointed to the target, e.g., $\overrightarrow{R_4}$ in Fig.6. If there are any obstacles in the direction to target (i.e. $\overrightarrow{R_4}$), then it is natural to choose $\overrightarrow{R_3}$ or $\overrightarrow{R_5}$ as the forward direction. If there are no obstacles in the directions of $\overrightarrow{R_3}$ and $\overrightarrow{R_5}$, then choose any one which is near to the robot's front end direction. If there are obstacles in the directions of $\overrightarrow{R_3}$ and $\overrightarrow{R_5}$, then choose $\overrightarrow{R_2}$ or $\overrightarrow{R_6}$ as the forward direction, and so on. The procedure is repeated in the same way until a feasible forward direction is found. If the robot is close to the obstacles, then a modified vector $\overrightarrow{f_m}$ is added to the forward direction so that the robot is kept away from the obstacles. The modified vector $\overrightarrow{f_m}$ is the summation of d_n, where d_n is function of $r_s - r_n$ and r_s is the safe distance. Note that the feasible forward direction must satisfy Eq.(8), or the heuristic approach fails and the tracking approach takes over.

In T-mode, the robot tracks along the obstacle boundary and keep a safe distance r_s from the obstacles. The forward direction is generated by the sensory feedback from the seven directions. The middle direction, i.e., $\overrightarrow{R_4}$ in Fig.6, is directed toward the front of the robot. As the track procedure is applied, the direction for tracking is determined from the orientation of the robot. When the robot tracks along the obstacle in a counterclockwise direction, the sensory feedback from the left four directions, i.e., r_4, r_5, r_6, r_7, are judged by several decision rules. Then the forward direction is decided. The rules are shown in Table 1. For clockwise direction tracking, decision rules have similar forms but with sensory data from r_4, r_3, r_2, r_1 directions. Note that weighting factors w_1 and w_2 have different values in each rule and their values are obtained by trial-and-error. There seems no systematic way to adjust them. Take the rule 1 in Table 1

as an example. If there are no obstacles in the front of the robot (i.e., $r_4 > r_v$) and there are obstacles in the left hand side (i.e., $r_6, r_7 < r_v$), then the robot should move in front. Hence, the first term of the desired forward direction $\overrightarrow{f_d}$ is $w_1 \overrightarrow{R_4}$. In order to keep a safe distance from the obstacle, $\overrightarrow{f_{r7}}$ is added as the second term, where $\overrightarrow{f_{r7}}$ is equal to $(r_s - r_7)\ \overrightarrow{R_7}$. The final forward direction is the combination of $\overrightarrow{f_d}$ and a modified vector which is the same as in H-mode. The other rules can be treated in the same way.

The rules used in the above seems very simple. However, the experiments show that these rules are quite robust and can tolerate a little erroneous measurement. Because the effective range r_v determines the reliable range of sensory feedback, r_v is an important parameter in the procedure. In H-mode, r_v depends on the environment. If the environment is crowded, r_v should be smaller. In T-mode, the tracking rules are taken for local decisions, r_v should not be large.

V. Computer Simulation and Implementation Results

The algorithm is confirmed to be feasible by simulation. In the simulation, the obstacle scattering is input from a CCD camera, and the range sensor is simulated by a straight line search method. Namely, we have assumed that the range sensor has perfect directional property and will not be affected by the distribution of obstacles. The simulation results are given in Fig.7 and Fig.8. The total simulation time are shown at the bottom line. The advancing distance in each step is 5 unit length. The total path length is represented as moving steps.

In Fig.7, since obstacles are all convex and small, only H-mode need to be used. It is intuitive that the path is shorter than that given by pure T-mode. In Fig.8, there are concave and convex obstacles. Both H-mode and T-mode should be used. The hit points and leave points are labeled and the switching between H-mode and T-mode is clear. Since the obstacle located at the center of the map is not large, H-mode is able to avoid it. In the simulation, the average computation time per step never exceeds 60ms. This means that the obstacle-free path can be generated in a control interval. Thus, the algorithm is feasible for real-time implementation. A complete simulation of different cases can be found in [8].

Next, the algorithm is actually implemented on our NTU II mobile robot. In the experiment, the obstacles are desks and wall in the laboratory. The scattering of the obstacles looks like an upside-down character U. The mobile robot has to avoid obstacles and safely reaches the target location. The straight line distance between the starting location S and the target location T is 270 cm. During the experiment, the sensory feedback data which are larger than r_v have been discarded. Since the obstacles are large relative to r_v, both the H-mode and T-mode are used in the experiment. The total path length of the obstacle-free path turns out to be 550 cm. The total traversal time of the mobile robot is twelve minutes. This value includes the sensor scanning time (470 seconds), the acceleration and deceleration times of motors. It is obvious that the use of one ultrasonic sensor is at the expense of long traversal time. Surprisingly, the position error at the target point is only 5 cm. This result shows the tracking ability of the robot. The experimental results are shown in Fig.9. Fig.9(a) shows the robot at the starting location. Fig.9(b) shows that the robot tracks along the obstacle boundary. Fig.9(c) shows the robot on the move. Fig.9(d) shows that the robot passes the other end of the obstacle. Fig.9(e) shows that the robot reaches the target location. The comparison between computer simulation and actual obstacle-free path is given in Fig.10. The actual path is drawn according to the inner position records of the robot. It is represented by circles. The obstacle-free path generated from the computer simulation is denoted by a solid line. The dots are sensory feedback data. Note that the two paths shown in Fig.10 are with different safe distance.

VI. Conclusion

In this paper, a feasible algorithm of obstacle avoidance is developed for the mobile robot system. The algorithm is based on sensory feedback without the help of any maps or mathematical models. The computing effort is small so that it can be real-time implemented. Though the mobile robot is fabricated by ourselves and only one ultrasonic sensor is used in the implementation, the tracking ability has been demonstrated by 5 cm tracking error. The implementation further justifies the feasibility of the combination of the heuristic mode and the track mode. The heuristic method has potential to be used in the environment with unmodeled multiple mobile robots. The decision error caused by the sensor error can be partially eliminated by using decision rules. If seven sensors are used to detect obstacles simultaneously, the total traversal time will be enormously reduced. In addition, if several types of sensors are used together, such as ultrasonic sensors, infrared sensors, and vision sensors, the reasoning on environment information will be more complete. The influence of sensor errors can be further reduced.

The proposed obstacle avoidance algorithm is able to find the obstacle-free path and can be physically implemented. However, the algorithm may not find an optimal path. If an optimal obstacle-free path is required, the model-based algorithm should be taken into account.

References

[1] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments," *IEEE International Conf. on Robotics and Automation*, Vol.1, pp.572-577, 1990.

[2] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots," *IEEE Trans. on SMC*, Vol.19, No.5, pp.1179-1187, 1989.

[3] R.A. Brooks, "Solving the Find-Path Problem by Good Representation of Free Space," *IEEE Trans. on SMC*, Vol.13, No.3, pp.190-197, 1983.

[4] D.F. Cahn and S.R. Philips, "ROBNAV: A Range-Based Robot Navigation and Obstacle Avoidance Algorithm," *IEEE Trans. on SMC*, Vol.5, No.9, pp.544-551, 1975.

[5] J.L. Crowley, "Dynamic World Modeling for an Intelligent Mobile Robot Using a Rotating Ultrasonic Ranging Device," *IEEE International Conf. on Robotics and Automation*, Vol.1, pp.128-135, 1985.

[6] A. Elfes, "A Sonar-Based Mapping and Navigation Systems," *IEEE International Conf. on Robotics and Automation*, Vol.2, pp.1151-1156, 1986.

[7] A.M. Flynn," Combining Sonar and Infrared Sensors for Mobile Robot Navigation," *International Journal of Robotics Research*, Vol.7, No.6, pp.5-14, 1988.

[8] H.P. Huang and P.C. Lee, "A Feasible Sensor-Based Algorithm for Obstacle Avoidance of Autonomous Mobile Robots," 14th National Symposium on Automatic Control, December 15, 1990, Hsinchu, Taiwan.

[9] S.S. Iyengar, C.C. Jorgensen, S.V.N. Rao, and C.R. Weisbun, "Robot Navigation Algorithms using Learned Spatial Graphs," *Robotica*, Vol.4, pp.93-100, 1986.

[10] S. Kambhampati and L.S. Davis, "Multiresolution Path Planning for Mobile Robots," *IEEE Journal of Robotics and Automation*, Vol.2, No.3, pp.135-145, 1986.

[11] D.M. Keirsey, E. Koch, J. McKisson, A.M. Meystel, and J.S.B. Mitchell, "Algorithm of Navigation for a Mobile Robot," *IEEE*

International Conf. on Robotics and Automation, Vol.1, pp.574-583, 1984.

[12] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *IEEE International Conf. on Robotics and Automation*, Vol.1, pp.500-505, 1985.

[13] D.T. Kuan, J.C. Zamiska and R.A. Brooks, "Natural Decomposition of Free Space for Path Planning," *IEEE International Conf. on Robotics and Automation*, Vol.1, pp.168-173, 1985.

[14] T. Lozáno-Pérez and M.A. Wesley, "An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles," *Communication of the ACM*, Vol.22, No.10, pp.560-570, 1979.

[15] V.J. Lumelsky and A.A. Stepnov, "Dynamic Path Planning for a Mobile Robot Automation with Limited Information on the Environment," *IEEE Trans. on Automatic Control*, Vol.31, No.11, pp.1058-1063, 1986.

[16] V. Lumelsky and T. Skewis, "Incorporating Range Sensing in the Robot Navigation Function," *IEEE Trans. on SMC*, Vol.20, No.5, pp.1058-1069, 1990.

[17] J.L. Olivier and F. Ozguner, "A Navigation Algorithm for an Intelligent Vehicle with a Laser Rangefinder," *IEEE International Conf. on Robotics and Automation*, Vol.2, pp.1145-1150, 1986.

[18] A. Sankaranarayanan and M.Vidyasagar, "A New Path Planning Algorithm for Moving a Point Object Amidst Unknown Obstacles In a Plane" *IEEE International Conf. on Robotics and Automation*, Vol.3, pp.1930-1936, 1990.

[19] E.K. Wong and K.S. Fu, "A Hierarchical-Orthogonal-Space Approach to Collision-Free Path Planning," *IEEE International Conf. on Robotics and Automation*, Vol.1, pp.506-511, 1985.

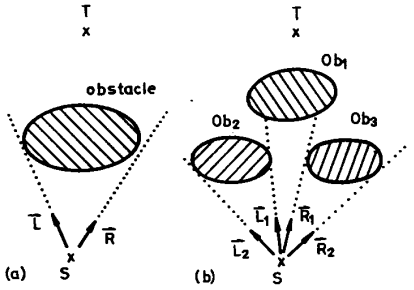

Fig.1 Forward directions of the robot.

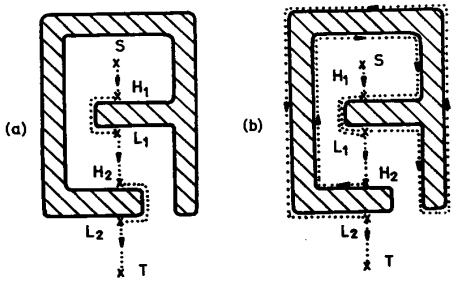

Fig.2 Solutions for avoiding a concave obstacle in T-mode.

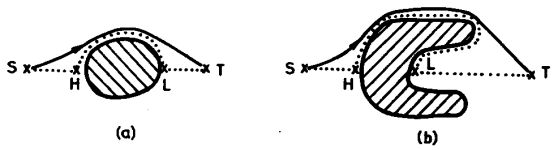

Fig.3 Comparison of H-mode and T-mode. The dash line denotes H-mode and the dot line denotes T-mode.

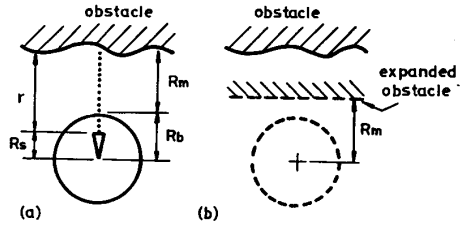

Fig.4 Modification of range data.

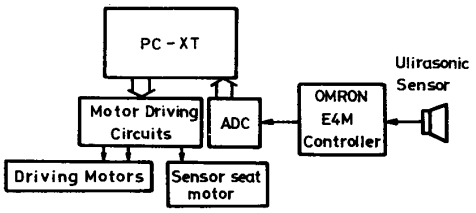

Fig.5 Block diagram of the robot system.

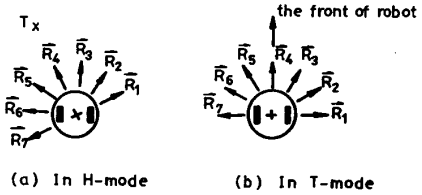

Fig.6 Directions for detecting obstacles.

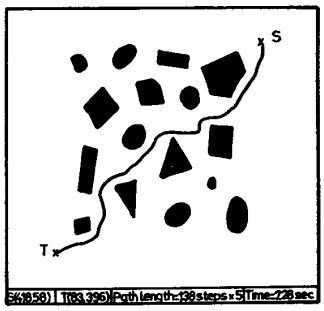

Fig.7 Simulation result, Case 1.

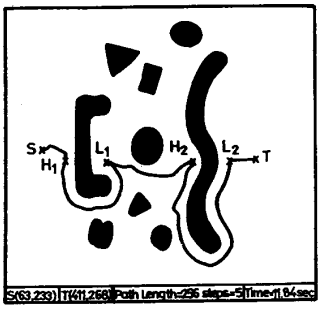

Fig.8 Simulation result, Case 2.

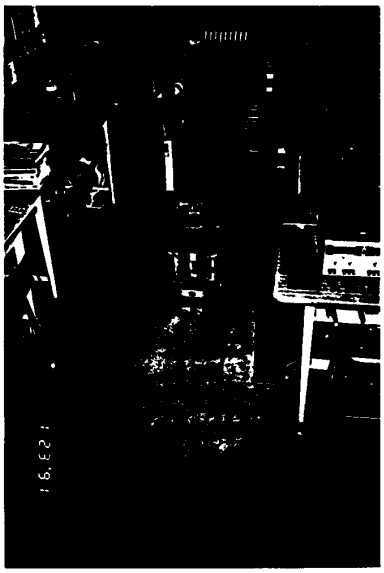

Fig.9(a) The robot at the starting location.

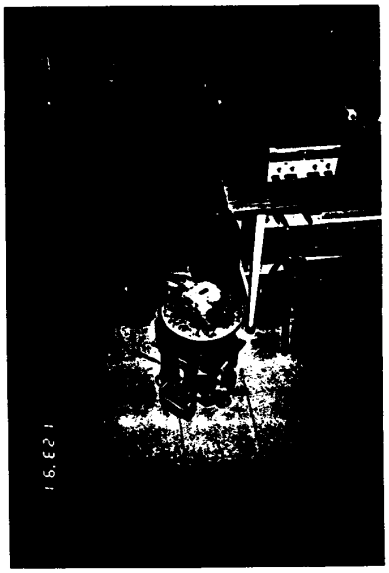

Fig.9(b) The robot tracks along the obstacle boundary.

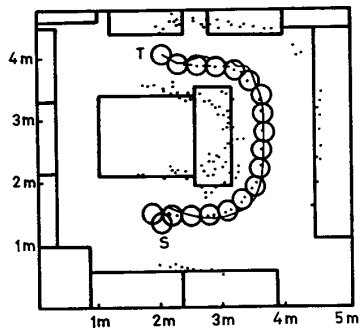

Fig.10 The comparison between computer simulation and actual obstacle-free path. The actual path is represented by circles and the computer simulation path is denoted by a solid line. The dots are sensory feedback data.

	r_7	r_6	r_5	r_4	Desired Motion	Resultant vector command
Rule 1	$< r_w$	$< r_w$	x	$> r_w$	Go in front	$w_1\vec{R}_4 + \vec{f}_{r7} + \vec{f}_{r6}$
Rule 2	$< r_w$	$> r_w$	x	$> r_w$	Turn left	$w_1\vec{R}_4 + w_2\vec{R}_7 + \vec{f}_{r7}$
Rule 3	$> r_w$	$< r_w$	x	$> r_w$	Turn right	$w_1\vec{R}_4 - w_2\vec{R}_7 + \vec{f}_{r6}$
Rule 4	$> r_w$	$> r_w$	$< r_w$	$> r_w$	Turn right	$w_1\vec{R}_4 - w_2\vec{R}_7 + \vec{f}_{r5}$
Rule 5	$< r_w$	$> r_w$	x	$< r_w$	Turn right	$w_1\vec{R}_4 - w_2\vec{R}_7 + \vec{f}_{r7}$
Rule 6	$< r_w$	$< r_w$	x	$< r_w$	Turn right	$w_1\vec{R}_4 - w_2\vec{R}_7 + \vec{f}_{r8} + \vec{f}_{r7}$
Rule 7	$> r_w$	$< r_w$	x	$< r_w$	Turn right	$w_1\vec{R}_4 - w_2\vec{R}_7 + \vec{f}_{r8}$
Rule 8	$> r_w$	$> r_w$	$< r_w$	$< r_w$	Turn right	$w_1\vec{R}_4 - w_2\vec{R}_7 + \vec{f}_{r5}$
Rule 9	$> r_w$	$> r_w$	$> r_w$	x	Turn left	$w_1\vec{R}_4 + w_2\vec{R}_7$

Table 1 Decision rules used in T-mode. The symbol "x" means do not care.