Fifth IFSA World Congress (1993), 1254-1257

Motion Planning of An Autonomous Mobile Robot in Flexible Manufacturing Systems

Yoo Seok Kim and Jang Gyu Lee

Automation and Systems Research Institute &
Department of Control and Instrumentation Engineering,
Seoul National University, Seoul, KOREA 151-742

ABSTRACT

Presented in this paper is a newly developed motion planning method of an autonomous mobile robot(AMR) which can be applied to flexibile manufacturing systems (FMS). The mobile robot is designed for transporting tools and workpieces between a set-up station and machines according to production schedules of the whole FMS. The proposed method is implemented based on an earlier developed real-time obstacle avoidance method which employs Kohonen network for pattern classification of sonar readings and fuzzy logic for local path planning. Particulary, a novel obstacle avoidance method for moving objects using a collision index, collision possibility measure, is described. Our method has been tested on the SNU mobile robot. The experimental results show that the robot successfully navigates to its target while avoiding moving objects.

I. INTRODUCTION

One of the most important and difficult problems in FMS, is the optimum supply of manufacturing equipment for optimal exploitation of capacity and independence[3]. One solution is developing flexible automated material handling and transport systems by using autonomous mobile robots. The mobile robot transports tools and workpieces to machines from depots according to the production schedules of the whole FMS. The robot in an industrial production environment must execute its tasks in a given time interval under the restrictions and requirements of a global optimum of the whole system.

It is impossible for the robot to estimate correctly trajectories of moving obstacles(robots) by using the information from ultrasonic sensors. At this time, if the workcell computer has information on the paths of those obstacles and can produce an advisable direction for obstacle avoidance, the robot autonomously avoids collision while trying to match the prescribed direction as closely as possible. Therefore, in order to maximize throughput of robot trips and avoid other moving objects, a new motion planning method combining reasoning, analyzing, and decision making of the workcell computer with an obstacle avoidance scheme has been developed. The core of this motion planning method is an avoiding algorithm for moving obstacles, which is executed by global controller, workcell computer. This motion planning is implemented based on an earlier developed obstacle avoidance method which employs the Kohonen network for pattern classification from ultrasonic sensor readings, and fuzzy logic for local path planning. An earlier obstacle avoidance method for static objects is briefly explained in section II and a new motion planning method is described in section III. Some experimental results with our two-wheel driven indoor robot are presented in section IV. The conclusions are discussed in section V.

II. REAL-TIME OBSTACLE AVOIDANCE METHOD

A distribution pattern of obstacles, which we call 'geometric characteristic' has to be drawn from a set of ultrasonic sensor readings. From these geometric characteristic, the robot assesses its surrounding situation and gets information on movement. The Kohonen network accomplishes pattern classification by making similar sensor reading sets clustered into the same category. Fig. 1 shows example of some classified patterns.

After a pattern classification, relatively safe steering directions for robot can be determined. According to the position of each sensor, steering sectors for each classified

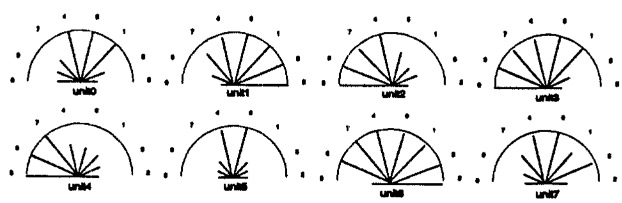

Fig. 1. Example of classified patterns.

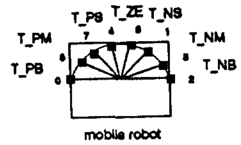

Fig. 2. Steering sectors.

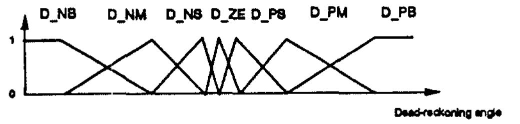

Fig. 3. Membership functions for dead-reckoning sector.

pattern are constructed as shown in Fig. 2. Employing the situation judgement based on the geometric characteristic, an intelligent control information can be obtained. The information is composed of two factors; the degree of relative safety for moving direction, and depth information of each sector. The information is assigned to each steering sector of all classified pattern via suitable linguistic value.

In order to navigate the robot to a designated target, the information about a route to target should be added to the fuzzy logic. Another fuzzy set called dead-reckoning sector is formed according to the direction pointing to the target. Fig. 3 shows the constructed dead-reckoning sector. A direction command is generated by a fuzzy logic which approximately reasons the two fuzzy sets using a suitable rule base. To test the performance of our method, a variety of experiments have been successfully performed with several local minimum traps[1]. Unlike the other methods, when robot is trapped, any trap-state detection scheme or other complementary recovery algorithms are not used.

III. MOTION PLANNING METHOD

A workcell computer controlling the flexible material handling and transport system has information on the paths of moving obstacles. The workcell computer coordinates the charging and discharging operation of mobile robot according to a production schedule. At the start, the robot moves using dead-reckoning sector representing the direction to the target. The workcell computer continuously monitors the trajectory of the mobile robot by means of

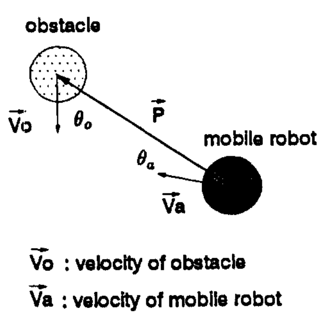

Fig. 4. Variables for defining collision index.

wireless modem. On navigation, when the robot encounters moving obstacles or confronted by difficult situation including the equlibrium state, the workcell computer can guide the robot similar to the teleautonomous operation[2].

The important problem is deciding the time when the workcell computer starts to coordinate motion planning to guide the robot. We define a collision index value at time t, denoted by $C(t)$, as

$$C(t) = \frac{|\vec{p}(t)|}{|\vec{v}_o(t)|\, cos\theta_o(t) + |\vec{v}_a(t)|\, cos\theta_a(t)} = \frac{|\vec{p}(t)|^2}{\vec{p}(t) \circ (\vec{v}_a(t) - \vec{v}_o(t))}$$

where $\vec{p}(t) = \vec{r}_o(t) - \vec{r}_a(t)$ and $\vec{r}_o(t)$, $\vec{r}_a(t)$ are position vectors of obstacle and mobile robot, respectively. Other variables are explained in the Fig. 4. The collision index value $C(t)$ indicates a collision possibility between obstacle and robot at time t. Because $C(t)$ has the time dimension, $C(t)$ can be considered as the collision expectation time.

On navigation, $C(t)$'s are always calculated for the robot and each obstacle by the workcell computer. If $C_i(t)$ is smaller than a presetted threshold value, C_t, avoiding algorithm of the workcell computer starts to coordinate robot motion. $C_i(t)$ is the collision index value between the robot and the ith obstacle and the smallest index value at time t. $C_i(t)$ is calculated by use of next position and next velocity of the robot and obstacle. Therefore $C_i(t)$ will be expectation value of collision index after one local control loop. The workcell computer searches a suitable next control value set consisting of forward speed and turning rate, which guarantees $C_i(t+1) > C_i(t)$ and minimizes a change of current robot speed. For each selected control value, each $C_j(t)$, $j \neq i$ is calculated to check whether $C_j(t+1)$ becomes smaller than C_t. If there are other obstacles whose next collision index value is smaller than C_t, cost value,

$$\frac{C_i(t+1) - C_i(t)}{C_i(t)} + \sum_{j \neq i} \frac{C_j(t+1) - C_j(t)}{C_j(t)}$$

is calculated for each selected control value and a control value set having the maximum cost value is finally found as output. Also, a virtual subgoal is generated by use of the current and next robot position and the target position.

The motion planning is implemented by replacing a dead-reckoning sector of the obstacle avoidance method with the general direction to the virtual subgoal position prescribed by the workcell computer. A direction command for the robot is generated by the fuzzy logic which has two input fuzzy sets, safety of steering sector and dead-reckoning sector representing the direction to the virtual subgoal. Our mobile robot is controlled by one control mode command called jog mode. The jog mode superimposes a continuous turn rate atop of existing forward speed. Turning rate is defined as an output fuzzy set. The most widely used max-min reasoning is employed in approximate reasoning. Considering a one dead-reckoning sector, each safety of seven steering sectors is approximately reasoned with it to produce the corresponding membership value of the resulting term in the turning rate. The crisp steering direction command is determined by the center of area, which performs with seven or fourteen direction components generated from the rule base.

Derivation of the rule base relies on experience and knowledge of mobile robot control. In particular, as described in the beginning of Section II, our work started on the assumption that a man assesses his surrounding situation by means of the distribution pattern of obstacles, and he acts in accordance with what that pattern might represent. The rule base is formulated employing a heuristic approach which involves an introspective verbalization of man. A deterministic method is used to derive the rule base that satisfies the control objective and constraint, i.e., robot's approaching a target and avoiding obstacles, respectively. The approximate reasoning determines membership functions of the turning rate, μ_A and jograte as follows.

$$\mu_A(\theta_n) = \max_{\phi_l, T\_m} \{\min[\mu_D(\phi_l), t(T\_m), \mu_R(\phi_l, T\_m, \theta_n)]\}$$

$$\mu_R(\phi_l, T\_m, \theta_n) = \max_{1\leq j\leq N} \{\min[\mu_{Dj}(\phi_l), t_j(T\_m), \mu_{Aj}(\theta_n)]\}$$

$$jograte = [\sum_k \sum_{i=1}^{7} \theta_i \mu_A(\theta_i)]/[\sum_k \sum_{i=1}^{7} \mu_A(\theta_i)]$$

Where T_m, ϕ_l represent a steering sector and a dead-reckoning angle, respectively. $t()$ and μ_D is denoted as safety of the steering sector and membership function of the dead-reckoning sector. The fuzzy relation R combines all the rules:

$$R = R_1 \cup R_2 \cup \cdots \cup R_N = \bigcup_{i=1}^{N} (D\_i \times T\_i \times A\_i)$$

N represents the number of rules and it is 49 in case of the above rule base. In the third equation, θ_i is the support value at which the membership function of the resulting output direction component reaches the maximum value. We call it vertex value. According to the dead-reckoning sector, the seven direction components are determined in the rule base.

A speed command is generated according to the depth information which is contained in the magnitude of safety value of steering sectors covering the resulting steering direction with respect to the turning rate. The suitability of steering sectors covering the steering direction is represented by a fuzzy set called speed sector. The speed component of speed sector term is represented by a fuzzy set called speed. In order to relate safety value of speed sector with speed component, the term set of safety also could be constructed the same as that of the speed fuzzy set. The determination of speed is made by simple calculations which only uses the fuzzification interface of linguistic variable. If the steering direction with respect to the turning rate is obtained, membership values for speed sectors are determined. According to safety value of each speed sector, the corresponding speed component is calculated. The speed command is generated from weighted sum of speed components and membership values of speed sectors. The detailed procedure can be found in [1].

TABLE I Experiment input data

Robot input data

initial position	(3000, 100)
heading	90°
target postion	(3000, 7000)
maximum speed	300mm/sec
maximum turning rate	100° /sec

moving obstacle's input data

obs. No.	initial posi. (mm)	direc. (deg)	speed(mm/sec)
1	(3000, 5000)	180	200
2	(7000, 6000)	130	310
3	(-2000, 5500)	250	150
4	(8000, 5000)	110	150

This motion planning has a shared control architecture. In shared control systems, a complex task is divided between the workcell computer and the mobile robot. In our motion planning system, the global control task of coordinating robot motion to avoid moving obstacles is performed by the workcell computer. This scheme plays an important role for robot to finish its task within a given time interval under production schedule. The local control task of guiding the mobile robot around unexpected static obstacles is performed by robot's onboard computer.

IV. EXPERIMENTS

In the experiments, four objects were detected moving toward the robot at different constant velocities and direction angles. Given that the robot is at the coordinate (3000,100) and has maximum speed of 300mm/sec, four obstacles are detected with the parameters, as is shown table I.

Fig 5.(a) shows location map for the robot and obstacles before global control action. In (b), the workcell computer guides the robot to avoid obstacle 1, 2. After the first global control, the robot is again guided to avoid obstacle 3, 4, as shown in (c). In (d), the robot navigates toward the target without help of global control. During the experiment, the robot navigates with maximum speed. Fig. 6 shows the SNU autonomous mobile robot.

V. CONCLUSION

A new motion planning similar to teleautonomous operation has been proposed for an autonomous mobile robot

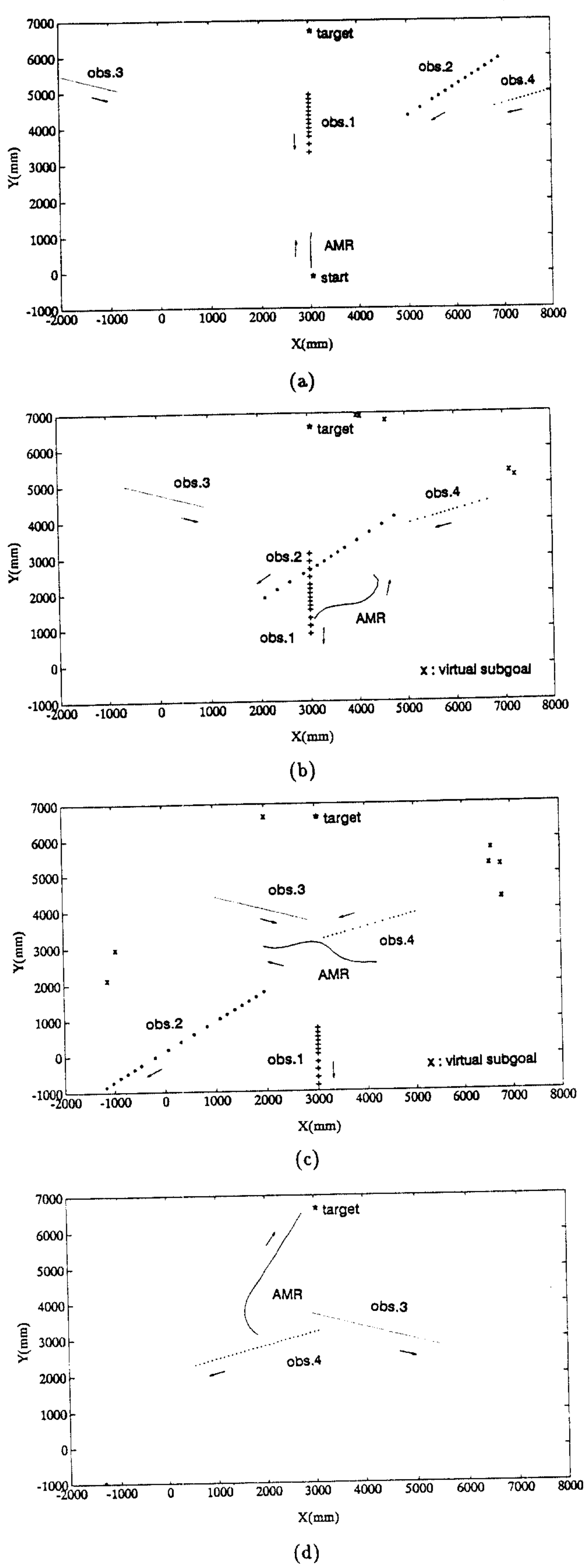

Fig. 5. Location map for the robot and four obstacles.

used in FMS. This method for the remote guidance of mobile robots uses a shared control architecture, in which the robot's on-site sensing and reflex capacity is combined with workcell computer's reasoning, analyzing, and decision making. When the robot encounters moving obstacles, the workcell computer guides it to avoid collision using the collision index and the search algorithm finding suitable control values. Our experiments show that the robot can navigate to a target avoiding moving obstacles.

ACKNOWLEDGMENT

This work has been partially supported by the Korea Science and Engineering Foundation Contract No. 90-1138, and the Ministry of Trade and Industry of Korea.

REFERENCES

[1] Y. S. Kim and J. G. Lee, "Robust Adaptive Control of An Autonomous Mobile Robot," *Proc. 1992 ICARCV Int. Conf. Automation, Robotics and Computer Vision*, vol.2, pp.INV-1.7.1-INV-1.7.5, 1992.

[2] J. Borenstein and Y. Koren, "Teleautonomous Guidance for Mobile Robots," *IEEE Trans. on System Man and Cybernetics*, vol.20, no.6, pp.1437-1441, 1990.

[3] J. Milberg and P. Lutz, "Integration of Autonomous Mobile Robots into the Industrial Production Environment," *Proc. 1987 IEEE Int. Conf. Robotics and Automation*, pp.1953-1959, 1987.

[4] T. David L. Brock, David, J. Montana, and Andrew Z. Ceranowicz," Coordination and Control of Multiple Autonomous Vehicles," *Proc. 1992 IEEE Int. Conf. on Robotics and Automation*, vol.3, pp.2725-2730, 1992.

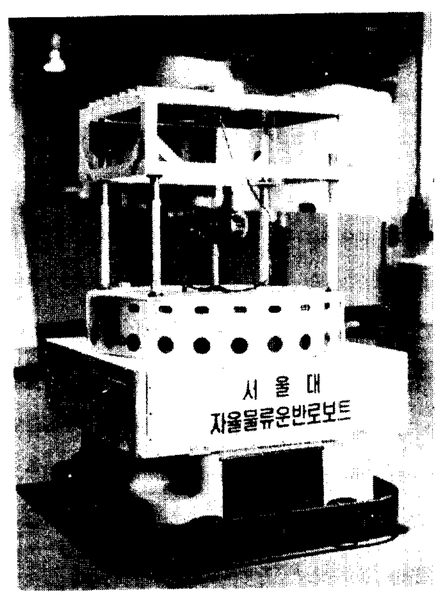

Fig. 6. SNU autonomous mobile robot.