Acquisition of Action for Mobile Robots Using Neural Network

Tomohiro Yamaguchi* and Yoshio Watanabe*

*Department of Electrical, Electronics and Information Engineering, Kanagawa University,
3-27-1 Rokkakubashi, Kanagawaku, Yokohama 221-8686, JAPAN
Email: tomohiro@ee.kanagawa-u.ac.jp

Abstract—Mobile robot path planning in a movement environment is an important problem. We studied acquisition of a path to a destination and obstacle avoidance of a mobile robot. The paper proposes a method of path planning based on neural network and genetic algorithm. The avoidance action of a mobile robot is determined from the obstacle configuration and the robot’s self-state using a neural network. The design parameter of neural network is adjusted by using genetic algorithm. The effectiveness of present method is proved through a simulation.

I. INTRODUCTION

Obstacle avoidance is one of the fundamental function of mobile robot. A mobile robot must be able to avoid both static and moving obstacles in its path. In the case where a robot cannot communicate with moving obstacles, the robot needs to predict the future motion of them. An obstacle avoidance of mobile robots with consideration to a moving obstacle is treated by some researches [1]-[3].

Moreover, two or more obstacles exist in the real-world. For this reason, obstacle avoidance of mobile robots is need to considering two or more obstacles.

This paper examines the multiple moving obstacle avoidance problem in a mobile robot. As a mobile robot, a wheel type robot with two independent drive wheels and one steering is used. For this reason, spin turn is included in action of a robot. In the obstacle avoidance problem of a wheel type robot, it is necessary to avoid all of obstacles by only turning.

Mobile robot actions, i.e., the velocity and turning angle, are determined from the obstacle configuration and the robot’s self-state using a neural network (NN), and multiple moving obstacles avoided. In order to avoid obstacles with the minimum movement time, the design parameters of the NN are optimized by genetic algorithm (GA), using the data collected from several environments, in which each environment has a different obstacle configuration. The effectiveness of present method is proved through a simulation.

II. ACTION DETERMINATION OF MOBILE ROBOT USING AN NN

An action of mobile robot is decided by a three-layered NN (Fig. 1). Input to the NN is assumed to be the position of obstacles \( \{x_{p_i}(k), y_{p_i}(k)\} \), the relative velocity of an obstacle to the robot \( \{x_{v_i}(k), y_{v_i}(k)\} \), the size of an obstacle \( s_{o_i}(k) \), and direction error, i.e., the direction between the destination and forward direction of the robot \( \theta_{de}(k) \). Output of the NN is the velocity of the robot \( \Delta v_r(k) \) and the turning angle of the robot \( \Delta \theta_r(k) \).

In this research, an action of the mobile robot to avoid multiple moving obstacles is provided by a three-layered NN shown in Fig. 2. Since the NN is changed with the number of obstacles, NN is prepared to each obstacle. In the input of NN shown in Fig. 2, Obstacle-1 is obstacle information that is the nearest position to a robot. Here, the maximum number of obstacle is set to 10 and the order of input information to NN is changed by near the position of obstacles.

A radial basis function neural network (RBFNN) [4], known as an NN that realizes various approximation functions, is used in the control system. With an RBFNN, a nonlinear function is expanded by any basis function having a circular contour, and is used as function approximation or pattern recognition. Unit functions at the hidden (or intermediate) layer of RBFNNs are given by

\[
\phi_{li}(x_l) = \exp \left\{ -\frac{\| x_l(k) - c_i \|^2}{\sigma_i^2} \right\}
\]

where \( \phi_{li} \) denotes ith unit output at the hidden layer of obstacle number \( l \), design parameters of RBF are center \( c_i \), and standard deviation \( \sigma_i \) for each input. jth unit output at output layer \( o_j \) is given by

\[
o_j(k) = \sum_{i=1}^{n} \sum_{l=1}^{m} w_{lij} \phi_{li}(x_l)
\]

calculated by a linear combination of outputs of the hidden layer. \( w_{lij} \) denotes the connection weight between the ith
hidden unit and the $j$th output unit of obstacle number $l$, and $m$ denotes the number of units at the hidden layer, where the number of units is determined by trial and error. The number of units was set to $m = 20$, because a good result was obtained when $m$ was three times the number of inputs.

Action of the mobile robot is determined from the information on multiple obstacles and the robot’s self-state using RBFNN.

III. ACQUISITION OF OBSTACLE AVOIDANCE BY SIMULATION

The simulation acquires action of multiple obstacles avoidance for mobile robot. The simulated environment is shown in Fig. 3, in which the $y$-axis is set to the forward direction of the robot. The robot is assumed to start from the center-of-gravity point of (0.0, 0.0) [m] and approach the goal, whose position is $y = 10$ [m]. Here, a robot’s size is set to 300 [mm] and the maximum velocity is set to 1.5 [m/s]. In this research, a robot has two independent drive wheels and one steering. For this reason, spin turn is included in action of a robot.

Moreover, the search range of the obstacles is shown in Fig. 4. Action of the mobile robot is determined using the information of obstacles that exists in the search range.

The block diagram of the present obstacle avoidance control system is shown in Fig. 5. RBFNN determines each amount of movements of the robot from the position of obstacles and robot’s self-state.

Obstacle information is updated from the amount of movements and the forward direction of the robot and obstacles.

The initial configuration of obstacles, i.e., position, velocity, forward direction and size, are determined at random. 100 kinds of environments with different initial configuration are prepared, and RBFNN is optimized. Moreover, the number of obstacle is set to 15 in one environment. Furthermore, obstacle is assumed to be going straight.

In simulation, connection weights of the NN and parameters (center and standard deviations) of RBFs are optimized by a GA [5] so that the robot avoids obstacles and reaches the destination with a minimum movement time. Table I shows design parameters for the GA used in simulation. The associated fitness function of an individual is defined by

$$fitness = \sum_{i=1}^{ob_n} (fitness_o + fitness_c)$$  (3)

whose solution is searched for as a minimization problem. $ob_n$ is the number of environments considered in optimization. $fitness_o$ is an evaluation function associated with penalty for
collision with an obstacle. \( \text{fitness}_o \) is given by

\[
\text{fitness}_o = \begin{cases} 
0.0, & \text{if there is no collision} \\
10.0, & \text{otherwise} \end{cases} \quad (4)
\]

Walking stops if the robot collides with an obstacle. \( \text{fitness}_c \) is an evaluation function related to movement time required to reach the destination, and given by

\[
\text{fitness}_c = T \times 10^{-3} \quad (5)
\]

\( T \) denotes movement time required to move from the starting point to the destination while avoiding obstacles. The maximum number of movement time \( T_{\text{max}} \) in one environment is set to 60 [s] and moving stops if movement time exceed \( T_{\text{max}} \).

IV. Simulation Result

The initial configurations of 15 obstacles in this simulation are shown in Table II. Here, \( xp \) and \( yp \) denotes the x- and y-directional coordinates of obstacles, \( vo \) denotes the velocity, \( \theta_0 \) denotes the forward direction, and \( so \) denotes the size. Figure 6 shows the initial position of obstacles and mobile robot. Here, the circle in Fig. 6 denotes the size of obstacles and mobile robot. The robot is start from the center-of-gravity point of (0.0, 0.0) [m] and approach the goal, whose position is \( y = 10 \) [m]. Fig. 7 shows the movement path of the mobile robot and multiple moving obstacles. The positions of obstacles and mobile robot after two seconds is shown in Fig. 7-(a), after four seconds shown in (b), after six seconds shown in (c), and after eight seconds shown in (d), respectively. Here, the arrow in Fig. 7 denotes the forward direction of moving obstacle. The mobile robot has avoided multiple moving obstacles. Velocity of the mobile robot for avoiding obstacles shown in Fig. 8. Furthermore, turning angle of the mobile robot for avoidance obstacles shown in Fig. 9. Moreover, the movement time to the goal is 8.1 [s].

V. Conclusions

In order to avoid multiple moving obstacles, the action of mobile robot has been determined through an RBFNN, whose inputs were the obstacle configurations and the robot’s self-state. Using the training data on several environmental conditions, the design parameters of the RBFNN were optimized using GA so that the obstacles could be avoided with the minimum movement time.

For the simulation result, we found that the RBFNN was useful for acquiring multiple moving obstacles avoidance action of mobile robot. However, since it realized in the decided environment, the obstacle of the environment that differed extremely may be unavoidable. Therefore, the effectness of the proposed system needs to be verified using the system.
Fig. 7. Movement path of the mobile robot and obstacles

Fig. 8. Velocity of mobile robot for avoiding obstacles

Fig. 9. Turning angle of mobile robot for avoiding obstacles

REFERENCES


