Improvement of path planning algorithm for mobile robot based on biologically inspired neural network

Shengmin Zhao  
Division of Electronics and Electrical Engineering  
Dongguk University  
Seoul, South Korea  
Shengmin.zhao@dgu.ac.kr

Seung-Hoon Hwang  
Division of Electronics and Electrical Engineering  
Dongguk University  
Seoul, South Korea  
shwang@dongguk.edu

Abstract—For path planning with full area coverage, the path planning algorithm based on the biologically inspired neural network has the problems of high traverse area repetition rate and large total traverse path length. In order to improve the path planning performance of mobile robots, this research improves this algorithm. First of all, in the escape algorithm, the method of real-time monitoring of the neuron state of the robot's neighborhood is adopted to shorten the escape path of the robot; Secondly, the neighborhood neuron state criterion is introduced, so that the robot first traverses along the edge of the obstacle when encountering an isolated obstacle to avoid obstacles. The simulation results show that the improved algorithm can effectively reduce the repetition rate of the traversed area, the total length of the traversed path, and the number of turns.

Index Terms—biologically inspired, path planning, neural network

I. INTRODUCTION

Mobile robots can replace humans in dangerous activities such as production, construction, and high wall cleaning. In the research of mobile robots, path planning is a focal issue. Most of the path planning of mobile robots is divided into two types. The first is point-to-point path planning and full-area coverage path planning. Point-to-point path planning, which requires the robot to find a path from the starting point to the endpoint in the working area [1]. The other is a complete traversal path planning, which requires the robot to find a path in the workspace that can traverse every feasible one in the environment Point path [2-3], so that the robot can traverse any passable point in the working environment when it ends. At the same time, the robot can avoid all obstacles autonomously and reach all areas except obstacles.

The path planning research of mobile robots generally includes two aspects: (1) the modeling method of the environment; (2) the path planning algorithm. The current modeling methods for the environment include template model method, topology method, visible method, free space method, geometric method, and grid method [4]. Path planning algorithms include genetic algorithm [5], ant colony algorithm [6], heuristic algorithm, A* algorithm [7], and so on. S. X. Yang et al. [8] applied the neural biologically inspired network model to the path planning of mobile robots and proposed an algorithm based on a biologically stimulated neural network, which has little dependence on the environment, even in environments with many obstacles. It can also traverse the work area well. But in this algorithm, when the mobile robot encounters an island obstacle or enters a dead zone, the planned path is longer. The mobile robot has a high traverse area repetition rate and a long traverse path.

This study improves the part of the escape algorithm through the method of real-time monitoring of the neuron state of the robot's neighborhood, and the introduction of the neighborhood neuron state criteria. The escape path of the robot is shortened, thereby reducing the total length of the planning path.

II. BASIC ALGORITHM

In the path planning algorithm based on biologically inspired neural network, the robot traversal path planning generation process is as follows:

Step 1: Calculate the activity value of each neighborhood neuron at the current position of the robot and take the maximum value.

Step 2: Determine whether the maximum value of neuron activity in the neighborhood is greater than the activity value of the neuron at the current position of the robot. If it is less than, the robot will fall into the dead zone. The robot should calculate the position of the neuron closest to the dead zone position and the activity value is greater than 0, then use it as the target point of escape. If it is greater than or equal to, the next step of the robot moves to the position of the neuron with the largest activity value among the neighboring neurons.

Step 3: Use the next move position of the robot as the new current position of the robot, and judge whether the traversal is completed. If the traversal is not completed, skip to step 1; if the traversal is completed, the algorithm ends.

III. IMPROVED ALGORITHM

A. Improved escape algorithm

When in the dead zone, the robot will first find the neuron on the map that is closest to the dead zone and whose activity value is greater than 0 and use it as the target point for escape. If the current position of the robot is consistent with the position of the escape target point, the algorithm ends. But because the escape target point obtained by the robot is not necessarily the optimal escape position, the escape path...
planned by the algorithm will be longer. After a lot of experiments, it is known that the optimal escape position of the robot is always on the path of the robot from the dead zone position to the escape target point. Therefore, when the robot moves to the target escape point, the state of the neurons in the neighborhood of the robot's current position can be detected in real time to determine whether it is escaped, so as to find the optimal escape position on the planned escape path, escape in time, and end escape algorithm.

The judgment condition of the improved escape algorithm is whether the cleaning robot reaches the target point of escape; the judgment condition of the improved escape algorithm is to detect whether there is a neuron with an activity value greater than 0 in the neighborhood. The improved escape algorithm allows the robot to detect the state of the neighboring neurons corresponding to the grid every time the robot moves to the escape target point.

**B. Improved obstacle avoidance algorithm**

According to the basic principle, the activity value of the neighborhood neuron determines the next move position of the robot. When the robot avoids obstacles, the path at A in Fig.1.(a) will appear. The robot will move in an oblique direction, resulting in a longer planned path. In order to shorten the path length, we stipulate that when the robot avoids obstacles, it first walks along the edge of the obstacle, and then executes the original algorithm. The ideal state of the obstacle avoidance planning path is shown in Fig.1.(b). Here we introduce the criterion of the state of the neighborhood neuron and stipulate that the criterion of the state of the neuron has a higher priority than the criterion of the maximum activity value of the neighborhood neuron. The robot first detects the distribution of each neuron state (obstacle, untraversed area, traversable area) of the neighboring neurons, and then determines the next action position.

Set a $30 \times 25$ grid environment map, each grid in the figure represents a neuron, set the starting point coordinates of the robot to $(2, 2)$, respectively use the biological excitation neural network algorithm and the improved The algorithm enables the robot to plan the path of the entire set environment. Fig.2(a) shows the planned path obtained by using the biological excitation neural network algorithm. Fig.2(b) shows the planned path obtained by the improved algorithm.

It can be seen intuitively from Fig.2(b) that the path obtained by the improved algorithm is shorter, the area of the robot's repeated traversal is reduced, and the obstacle avoidance planning performance is better. It can be seen intuitively from Figure 5 that the path obtained by the improved algorithm is shorter, the area of the robot's repeated traversal is reduced, and the obstacle avoidance planning performance is better.

**IV. CONCLUSION**

In this paper, the path planning algorithm based on biologically stimulated neural network is improved: real-time monitoring of the neuron state of the robot's neighborhood is used to determine whether it is out of trouble, so as to shorten the robot's escape path. The neighborhood neuron state criterion is adopted to make the robot traverse along the edge of the obstacle first when encountering an isolated island obstacle to avoid obstacles. The algorithm before and after the improvement is simulated. The improved algorithm can reduce the repetition rate of the traverse area, the total length of the traverse path and the number of turns on the premise that the robot traversal completion rate is 100%.

**REFERENCES**


