Obstacle Avoidance in Mobile Robot using Neural Network

Kai-Hui Chi
Graduate Institute of Automation and Control
National Taiwan of University Science of Technology
Taipei, Taiwan
M9812210@ mail.ntust.edu.tw

Min-Fan Ricky Lee
Graduate Institute of Automation and Control
National Taiwan of University Science of Technology
Taipei, Taiwan
rickylee@mail.ntust.edu.tw

***Abstract*—Investigate mobile robot's history, obstacle avoidance is one of most important research area and also the foundation of building robot's successful behaviors. This paper proposes a Neural Network control system that is able to guide the mobile robots (AmigoBot and P3DX) traverse through a maze with arbitrary obstacles. The pattern is trained by using Matlab toolbox and Aria library for motion control. There are 256 specific patterns defined to help robot organize the situation. For input data, sonar and laser range finder are two main sensors for passing on information of environment. The empirical results show the effectiveness and the validity of the obstacle avoidance behavior of Neural Network control strategy.**

Keywords- Mobile Robot; Intelligent Control; Neural Network; Obstacle Avoidance.

I. INTRODUCTION

Mobile robot has been used for various purposes in many application fields including exploration, industry etc. The uncertainties in sensors and the environment affect the obstacle avoidance algorithms during navigation.

The mobile robot system can sense environment from various sensors (e.g. sonar, laser range finder, IR, and CCD etc.). Beom and Cho [1] used sonar to simulate the Neural Network patterns to judge the situation. After ensure the acting model, respond by using Fuzzy logic to control motion and implement obstacle avoidance successfully. Demirli [2] also execute sonar data can be used as input to fuzzy sets for the global mapping that is to model environment and indentify robot's position and orientation. Make the good accuracy and performance of sonar data detection. On other case, Ganapathy [3] concluded that his proposed behaviors of the controllers with Path Remembering make the mobile robot to be capable in reaching the desired goal by Artificial Neural Network.

Harb [4] used Neural Networks to recognize environmental recognition and control speed of the mobile robot using Fuzzy Logic system to guide a mobile robot to track a predefined path to arrive at final goal.

Jensfelt [5] proposed a Kalman filter-based approach using laser scanning and a minimalistic environmental model. They demonstrated a low-complexity algorithm with a high degree of robustness and accuracy.

Lin [6] provides fuzzy extended information filter (FEIF) to implement robot's posture estimation and track autonomous mobile robot with odometer's information. Simulation and experiment makes efficacy results and useful of proposed method in [6].

Motlagh et al. [7] demonstrated that Fuzzy Logic systems can tolerate uncertain and imprecise information using linguistic rules. Main tasks are switched strategy of obstacle avoidance, target seeking and actual–virtual target to resolves the problem of any dead-ends encountered on the trajectory to the target. Simulation result demonstrates the trajectories are effective in related method.

Chang [8] simulated the motion planning using a scanning laser range finder and camera. Approached methods are varied three different mode of wall following, normal and sub-target in the process of experiment. Robot adjusts the mode of sub-target to reach the goal and conquer the problem of dead lock situation. Simulation results show the robustness of the proposed work.

A considerable amount of literature has been published on obstacle avoidance, dynamic environment recognition and mobile robot speed control using Fuzzy Logic and Neural Network [1-4]. We applied the methods in [1] as main approached structure on two mobile robot platforms, AmigoBot and P3DX, and further extend several more conditions from the previous researches.

This paper proposes an intelligent control system applying artificial neural network to learn the environment from the sonar sensory data to navigate the mobile robot to move along a collision free trajectory. The intelligent control system is tested in the mobile robots (Amigo and P3DX) under unknown environment.

256 specific patterns from 8 sonar reading model the entire possible environment scenario and off-line trained and learned by the artificial neural network to compare with Beom and Cho's work [1] using laser range finders. In this paper, one approach of method is that only using Neural Network to organize any situation and respond as training result.

The contribution of the propose work is no implicit modeling of mobile robot and environment is required to reduce the computation for real-time navigation. The Neural Network algorithm in this paper exerts a powerful effect upon obstacle avoidance.

II. METHODOLOGY

The two mobile robot platforms P3DX and Amigo (ADEPT TECHNOLOGY INC) with eight sonar sensors on the front are used as the mobile robot platform shown in Figure 1.

Figure 1. (Left) P3DX (Right) Amigo

The 8 sonar data as shown in Figure 2 is not reliable due to the phantom effect in the proposed maze environment.

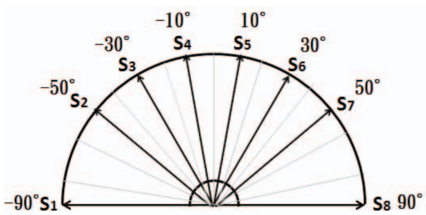

Figure 2. sonar direction and degree in P3DX

In alternative the laser range finder (Hokuyo URG-04LX) is used that its 681 reading divided into 18 regions (10 degree span) as shown in Figure 3. The data randomly lost (zero reading) as the blank part shown in Figure 3. The neighborhood region reading is averaged to fill those invalid data.

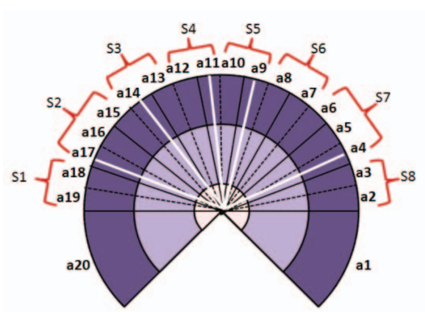

Figure 3. Configuration from a1-a19 to s1-s8 and blank present data lost

Eighteen regions (a2-a19) are then further grouped into eight areas (s1-s8) as shown in Figure 3. In a sensing area of Laser range finder, the radius 700 mm is setting for the data sensing range. In training of Neural Network, rule defines the radius 700 mm as threshold for detecting obstacle. The trigger will turn to one (1) when the obstacle is sensed under threshold. Otherwise, zero (0) will presents while nothing is detected under a distance of threshold. In example of Figure 4, input will present as [0 0 0 0 0 0 1 0]. Then pattern will implement the trained result to suit real time situation.

We complete the data training from simulations of obstacle position. In rule definition, we make 8 regions to organize the situation of the environment. There are 2 different states of 1 & 0 (Trigger by threshold), so the result of training will have 2^8=256 patterns to help robot can handle the unknown environment. For example, if an obstacle is positioned as shown in Figure 4, the input is [0 0 0 0 0 0 1 0].

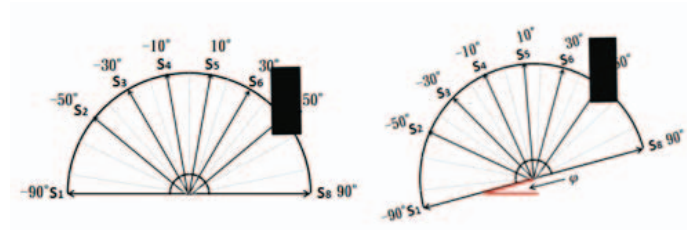

Figure 4. (Left) Obstacle Checked (Right) Amigo Respond of mobile robot

In this case of example, the trained result of algorithm will execute output ϕ= 16 degree (turn left 16 degree) as showed in Figure 4. For data training, we construct all obstacle possibilities and define respond of the output.

The Back propagation model is used in Neural Network methodology to train data. Several learning rate update methods are implemented through experiments and the performance is analyzed to achieve the highest pattern recognition rate with smallest error, e.g., gradient descend, gradient descend with adaptive learning rate, Levenberg-Marquardt and Fletcher-Reeves. The Back propagation model [9] is summarized as

Step 1 Initialize weights and thresholds to small random values.

Step 2 Choose an input-output training data set ($x(k)$,$t(k)$)

Step 3 Compute NN signals from input to output:

$$O_p^{(l)} = f\left(\textstyle\sum_i W_{pi}^{(l)} O_i^{(l-1)}\right) \tag{1}$$

Step 4 Compute output error E and Back propagation parameter at the output layer (L)

$$\delta_i^{(L)} = [t_i - o_i^L][f'(tot_i^{(L)})] \tag{2}$$

$$E = \tfrac{1}{2}\textstyle\sum_{i=1}^{n}(t_i - o_i^l)^2 + E \tag{3}$$

Step 5 Update the weights using:

$$\Delta W_{if}^{(l)} = \eta \delta_i^{(l)} o_j^{(l-1)} \tag{4}$$

With Back propagation:

$$\delta_i^{(l-1)} = f'\left(tot_i^{(l-1)}\right) \textstyle\sum_R \delta_p^{(l)} w_{pi}^{(l)} \text{ form } l = L \text{ to } 2 \tag{5}$$

Step 6 Repeat steps 2-5 for another training data set and compute error.

Step 7 After using all training data sets (i.e., one epoch), if final error E is less than a predetermined tolerance. The network has been trained. If not, repeat the process for another epoch.

Where output layer $\delta_i^{(l)} = -\frac{\partial E}{\partial o_i^{(l)}} f'(tot_i^{(l)})$ and Sigmoidal function (with λ= 1): f "= f(1-f)

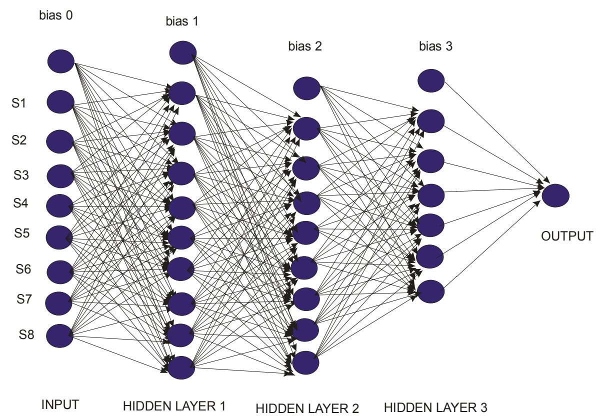

Figure 5. The main Neural Network structure

First, we make new network. Then we set parameters, such as how many iterations in simulation, Mean Square Error (MSE), how fast simulation learn the pattern of relationship between input and target, and displayed every 100 iterations until goal is reached that is the training error converged within the tolerance. The final result of simulation is that we get the weight values connecting among input to hidden layers and output.

After we try several algorithm methods and hidden layer amount combination, we decide that combination with 9 neurons in hidden layers 1, 8 neurons in hidden layer 2, and 6 neurons in hidden layer 3 are the Neural Network structure we use. Figure 5 is main structure of Neural Network algorithm.

TABLE I. PATTERN OF NEURAL NETWORK(UNIT: DEGREE)

Pattern Num	s1	S2	s3	s4	s5	s6	s7	s8	result	target
No.1	0	0	0	0	0	0	0	0	-0.3728	0
No.2	1	0	0	0	0	0	0	0	-2.0152	0
No.3	1	0	0	0	0	0	0	1	-0.2938	0
...	...	...	...	...	...	...	...	...	...	...
No.17	0	0	0	0	0	1	0	0	-15.7171	-16
...	...	...	...	...	...	...	...	...	...	...
No.58	1	0	0	0	1	0	0	1	-35.4436	-36
...	...	...	...	...	...	...	...	...	...	...
No.114	1	0	1	0	1	0	0	0	58.3856	58
...	...	...	...	...	...	...	...	...	...	...
No.242	1	0	1	1	1	1	0	1	45.153	45
...	...	...	...	...	...	...	...	...	...	...
No.254	1	1	1	1	1	1	1	1	86.188	87.1
No.255	1	1	1	1	1	0	1	1	86.5298	87.2
No.256	1	1	1	1	0	1	1	1	86.5633	87.3

Fig5 EX

After the structure established, the system can recognize the pattern which is trained. Table I shows the 256 patterns and training results. In observation of Table I, positive sign denotes for turning right and minus sign denotes for turning left. The patterns are represented by the combination of 0 and 1 and each pattern corresponds to a turning angle. Figure 6 shows the relationship of NN output vs. previously defined target.

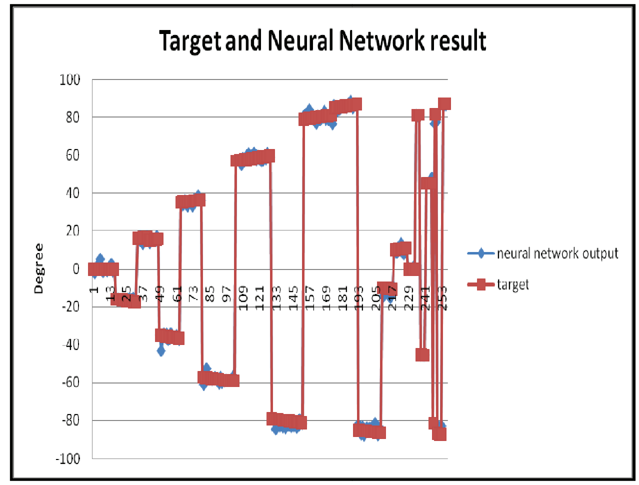

Figure 6. Target output and Neural Network output

Figure 7 shows the flow chart of the close loop system algorithm. The indoor environment is sensed first. When valid data arrived, the regions separated as the different areas and continue the primary algorithm computing of Neural Network system. Upon the various environment the system encountered and recognized, corresponding decision making on heading angle is input to robot to avoid hitting obstacle and pass through the maze.

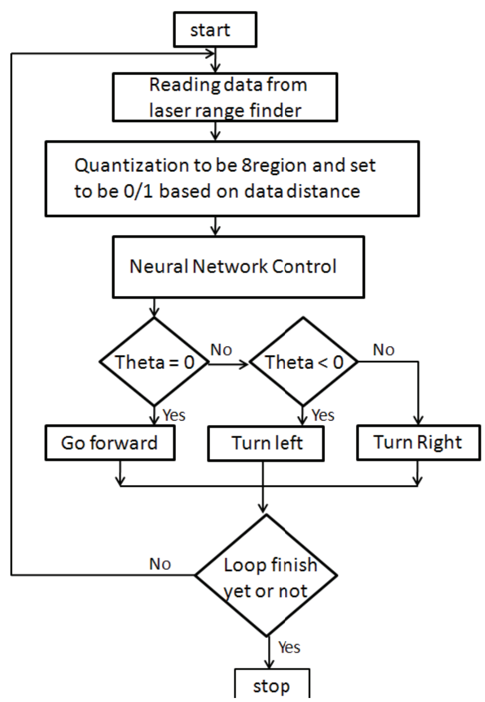

Figure 7. Flow chart of main algorithm structure

III. RESULT

The environment of Neural Network is applied to P3DX mobile robot for search way to exit from the maze path as shown in Figure 8

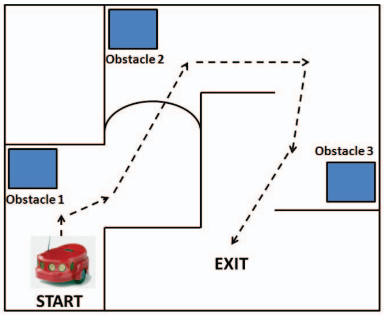

Figure 8. Application of maze

The sensitivity is up to how region designed. When distance of definition is too far, it will cause system more sensitive and robot's movement became unstable. On the contrary, if sensing range is too close, sensitivity decreased and the mobile robot has a higher chance hitting the obstacles. From experiment observation, the appropriate distance for the region which is 700 mm as shown in Figure 9. This distance is suitable for the maze in undefined environment.

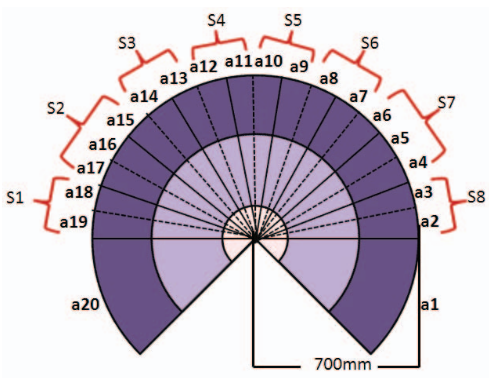

Figure 9. The sensing range of Neural Network

Figure 10 shows the trajectory is recorded by a ceiling mounted omnidirectional to evaluate the performance of the proposed system traverse through the maze while avoiding the multiple obstacles. The mobile robot successfully planed a collision free trajectory in the maze. The trajectory also shows the stability and accuracy of the proposed Neural Network Control algorithm.

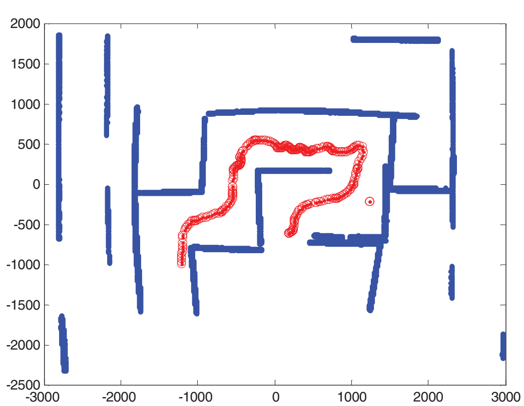

Figure 10. Trajectories and maze plot(The unit of x-y label is mm)

IV. CONCLUSIONS

In this paper, we multiple rules are implemented for the control strategy to avoid the obstacle successfully. The different arrangement combinations which based on *"0"* or *"1"* in a region construct a total 256 patterns. Laser range finder is used to sense the environment. Although laser ranger finder also sometimes has error data, it is eliminated by further filtering for non-zero regions. The proposed system in this paper with Neural Network control approach has demonstrated the effectiveness on avoiding the obstacles and robot can navigate through the trajectory with stability and reliability. In summary, the proposed system proceed as follows,

1. System takes data from sensors (laser range finder or Sonar). It is a quantization to be eight regions for Neural Network control.
2. The reading data will be condition for decision making in controller (Neural Network).
3. Output from controller will be used to command mobile robot's heading direction. If the decision on heading degree is not equal to zero, mobile robot will implement action based on algorithm which is designed to translate and/or rotate to avoid the obstacle.

To achieve better response from the proposed Neural Network approach, it needs further experimental investigation to have better data training. The more valid trained data, the better Neural Network system we have. For the application in real world, this research can be used in homecare for elderly and disabled as mobile robot navigating in the complex environment.

REFERENCES

[1] Beom, H. R. and H. S. Cho (1992). "A Sensor-based Obstacle Avoidance Controller For A Mobile Robot Using Fuzzy Logic And Neural Network." Intelligent Robots and Systems, 1992., Proceedings of the 1992 IEEE/RSJ International Conference.

[2] Demirli, K. and I. B. T¸rksen (2000). "Sonar based mobile robot localization by using fuzzy triangulation." Robotics and Autonomous Systems 33(2-3): 109-123.

[3] Ganapathy, V., Y. Soh Chin, et al. (2009). "Fuzzy and Neural controllers for acute obstacle avoidance in mobile robot navigation." Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference.

[4] Harb, M., R. Abielmona, et al. (2009). "Speed control of a mobile robot using neural networks and fuzzy logic." Neural Networks, 2009. IJCNN 2009. International Joint Conference.

[5] Jensfelt, P. and H. I. Christensen (2001). "Pose tracking using laser scanning and minimalistic environmental models." Robotics and Automation, IEEE Transactions on 17(2): 138-147.

[6] Lin, H. H. and C. C. Tsai (2008). "Laser pose estimation and tracking using fuzzy extended information filtering for an autonomous mobile robot." Journal of Intelligent and Robotic Systems: Theory and Applications 53(2): 119-143.

[7] Motlagh, O. R. E., T. S. Hong, et al. (2009). "Development of a new minimum avoidance system for a behavior-based mobile robot." Fuzzy Sets and Systems 160(13): 1929-1946.

[8] Yau-Zen, C., H. Ren-Ping, et al. (2007). "A Simple Fuzzy Motion Planning Strategy for Autonomous Mobile Robots." Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE.

[9] Karray, F.O. and de Silva, C.W., Soft Computing and Intelligent Systems Design—Theory, Tools, and Applications, Addison Wesley, Harlow, U.K., 2004.