Feed-forward Neural Network approach in Navigation control and P-I speed control of Mobile Robot

[1]Jeffina Stephens S, [2]Maheshan C M, [3]Prasanna Kumar H

[1]PG Scholar, [2,3]Asst. Professor, Department of Electrical Engineering
[1,2]UVCE, Bangalore, India

Abstract: **In this article, the implementation of Artificial Neural Network (ANN) approach for obstacle avoidance in Mobile Robots is presented. A basic method to interpret the data from the proximity sensors and a single layer feed forward Neural Network algorithm for the navigation control is proposed. The proximity sensor array outputs are arranged in combinations of '0' and '1'. This strategy allows training the Neural Network using only one type of sensor to detect the presence of obstacle. A suitable algorithm is applied for adjusting the network weights. User directed the robot through a GUI. In addition to this, the speed control of Mobile robot is achieved by a Proportional- Integral controller. The trained neural network has been tested with the real sensor data. The robot has been wirelessly connected with the Neural Network controller. Experimental results validate the effectiveness of the proposed approach in mobile robot navigation. The training of Neural Network is carried out by using MATLAB platform.**

Keywords: **Artificial Neural Network; Navigation Control; PI; Mobile Robot; MATLAB.**

I. INTRODUCTION

Mobile Robots have been the topic of research owing to its flexibility and wide applications. Mobile Robot can be defined as a robot system that is capable of traversing in an environment with unguided pathway and surrounded by unknown obstacles [1].The flexibility of a mobile robot is because of its ability to execute many tasks that are considered hazardous or tedious for humans [2]. For robots to achieve intelligence and autonomous navigation, precise control paradigms need to be developed.

The intelligence and self-directed control system can be built on robot using techniques such as Artificial Neural Network, Fuzzy Logic etc. This appends a decision making quality on to the robot system. The ability to learn nonlinear relationships between the inputs and the outputs makes the above said techniques suitable for real world problems. The similar kind of works by other authors that inspired the making of the proposed work is mentioned.

Timothy [2] in his work used simulated sensor data for the NN training. The user selects the operation which was then used to train the NN. The Extended Delta-Bar-Delta (EDBD) algorithm trains the network which allows the synaptic weights to have independent learning rate. Medina- Santiago [3] uses ultrasonic sensors for obstacle detection. An embedded platform based on ANN has been used. The pattern classification is dealt with Multilayer Perceptron.

Kai-Hui Chi and Min-Fan Ricky Lee [4] demonstrates the effectiveness of Neural Network control strategy by applying it to learn the environment from the laser range sensor data to traverse the Mobile Robot along a collision free path. Error elimination is achieved using filtering. Mustafa I. Hamzah and Turki Y. Abdalla [5] in their work introduce a navigation scheme using a combination of fuzzy logic and wavelet network. A goal reaching fuzzy logic algorithm and kinematic model is built and tested.

In the proposed study, navigation control is achieved using ANN. An array of Infra-Red (IR) proximity sensors are used to detect obstacles, which provides the input pattern to the NN controller. MATLAB environment is used for ANN training. The trained pattern is identified with the input from the sensor time to time and accordingly the output is communicated. Proportional integral controller governs the speed control of the mobile robot. A suitable hardware model is used to prove the practical effectiveness of the proposed study.

II. PROPOSED METHODOLOGY

A. Mechanical design and Arrangement

With focus on ruggedness and future expansion, the mobile robot has been designed to be functionally stable and structurally small (Fig. 1). The body of the mobile robot was built using Hylam sheets that can tolerate varying pressure applied on it. The proximity sensor array is bolted to the front end of the mobile robot. The motors are connected to the rear end of the robot. The motors used are DC gear motors. The dimensions of the robot are: 37cm × 25 cm.

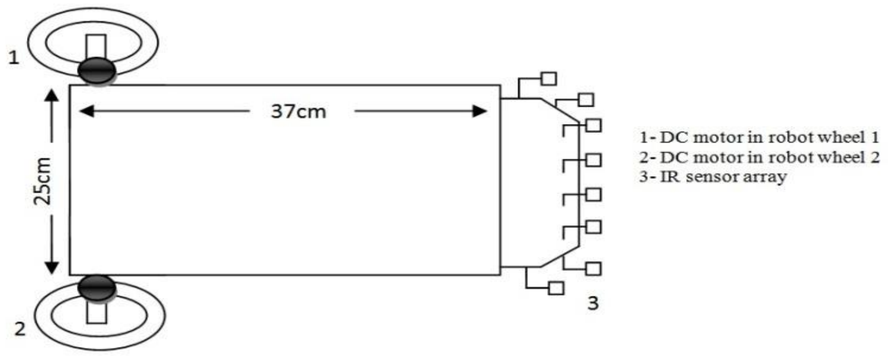

Fig. 1: Mechanical Design and Arrangements

B. Instrumentation Aspects

The navigation is achieved by avoiding the obstacle in the way of the robot and traversing towards the target. The detection of the obstacle is facilitated with the Infra-Red (IR) proximity sensors that serve as the eyes of the system. To achieve speed control, the speed of the motor is detected using motor speed sensor working on photo-electric effect.

The IR proximity sensor provides logic high to the controller when an obstacle is detected, thus providing input to the NN controller. The characteristic feature exhibited by the sensor is shown in Fig.2. It can be inferred that as the distance increases, the output voltage decreases. The sensitivity is found to be 0.2V/cm. The speed of the motor connected to the robot's wheel is measured using wheel encoder. The sensor works on the photoelectric principle, consists of an IR LED and phototransistor placed on opposite sides. A disc with spokes is attached to the shaft of the motor and rotates in between. The number of light interruptions is counted by the microcontroller and additional circuitry.

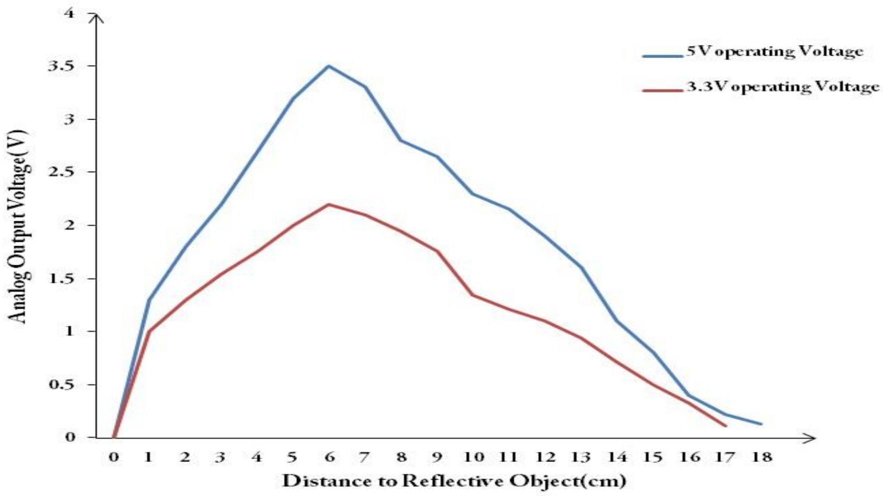

Fig. 2: IR sensor characteristic curve

The digital control aspect is facilitated with a PIC microcontroller, which has an operating frequency of about 20 MHz On the control board was attached a CC2500 RF Module, serial modem for communicating signals between MATLAB and the microcontroller. Additionally, a motor driver is used for the signals from the microcontroller to the motor, meet the necessary current specifications.

C. System's Functional flow

The functional flow of the proposed project is shown in Fig.3. The IR sensor array, in the presence or absence of an obstacle sends signal to the controller, which in turn is communicated to the NN controller. With supervised learning applied, the input and output pattern is trained and stored. When the input is received from the controller, it is compared with the stored pattern and the best fitting output, which is the angle of avoidance, is communicated back to the controller. The controller sends the suitable signal to the DC motors connected to the wheel of the Mobile Robot. The Proportional Integral algorithm is embedded in the microcontroller. Based on the output from the wheel encoder, the necessary control action is achieved through the controller.

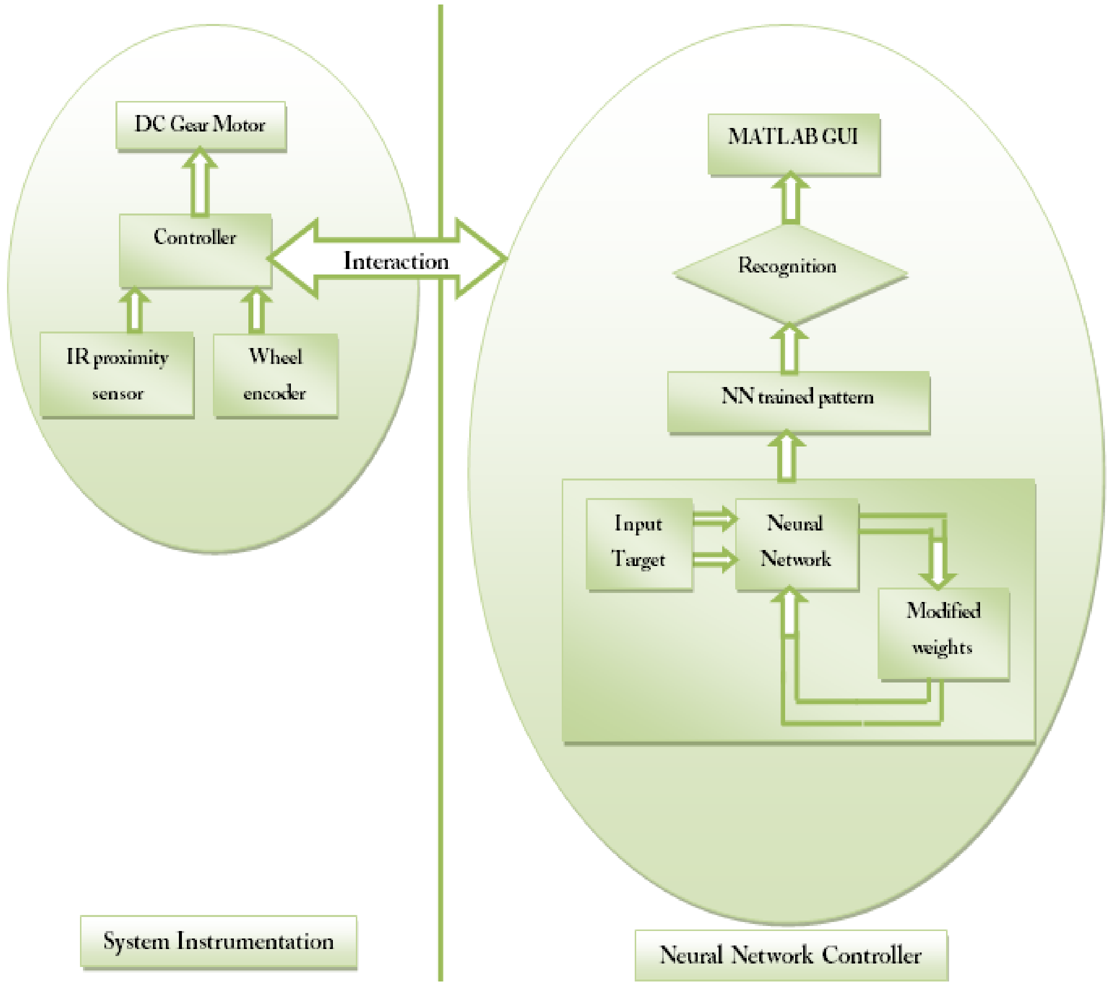

Fig. 3: Functional Flow chart of the system

III. MOBILE ROBOT SPEED CONTROL

A. Proportional Integral (PI) controller in motor speed control

It is very essential to have a control over the speed of the motor as it determines the speed/motion of the entire robotic system. Considering the dynamical nature of the system, a proportional integral control strategy is applied [5]. With PI control, the steady state accuracy can be improved. From the simulation results (Fig. 4), it can be inferred that the PI control is more suitable for the proposed system as compared to Proportional Integral Derivative control.

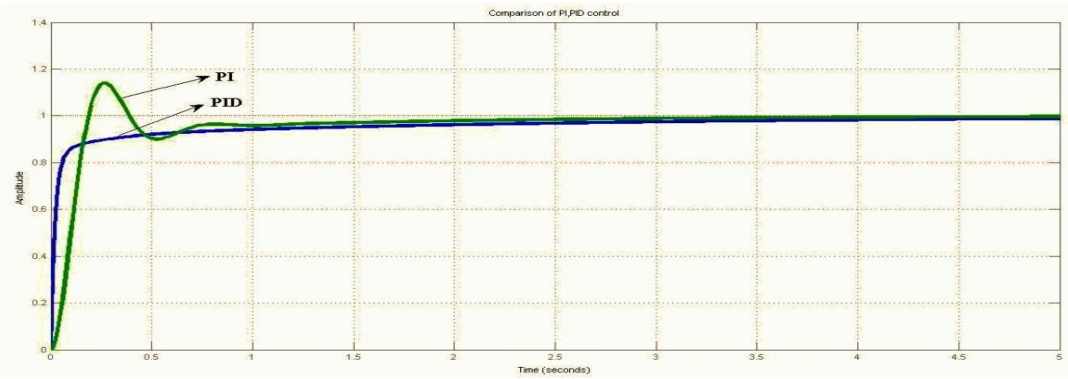

Fig. 4: Comparison between PI and PID

IV. ANN CONTROLLER DESIGN

The subject of ANN has achieved drastic significance over the last few decades. Neural Network derives its origin from the human nervous system consisting of incredulously prominent interconnection of large quantities of neurons that performs various perceptual tasks in minimal time as compared to the fastest computers. Artificial Neural Network controller is developed using MATLAB. A GUI is also developed for the user control.

A. Supervised learning

The cognitive process involved is supervised learning, wherein the algorithm is fed with the examples of input and output that is desired to be computed. The output from the IR proximity sensor serves as the input for the NN training. The sensor output is ‘1’ or ‘0’. There are eight sensors providing input to the network. Thus, 2^8 =256 combinations of input serve in detecting obstacles. For instance, the input array to the NN when sensor 4 and 5 goes high is [0 0 0 1 1 0 0 0].

B. Back propagation (BP) algorithm

The BP algorithm uses the supervised learning [7]. The BP model [7] [8] is used in layered feed-forward ANN.

The artificial neurons are stacked as layers. The signals are sent forward; the error is computed and propagated backwards. The algorithm is provided with samples of inputs and outputs, error is computed. The principle of BP algorithm is to reduce the error. The learning process starts with random synaptic weights.

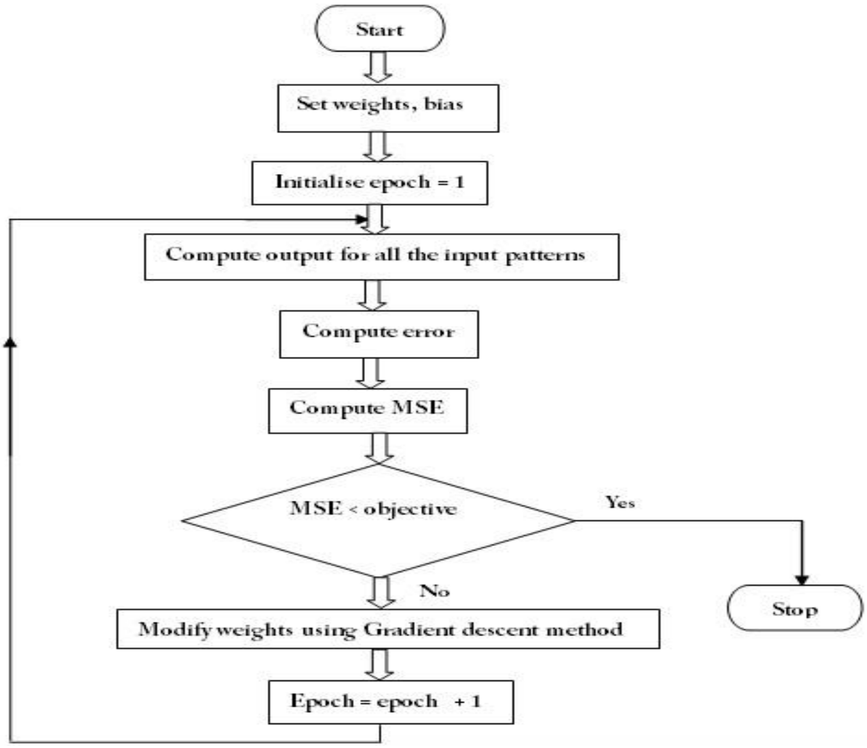

Fig.5: Flow chart for Back propagation method

International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 4, Issue 3, pp: (95-101), Month: July - September 2016, Available at: www.researchpublish.com

D. Adapted Neural Network architecture

The NN architecture is applied in the MATLAB environment. The number of neurons in the input layer is eight. Each neuron corresponds to one among the eight IR proximity sensors connected to the system. The output of the NN is the angle by which the system should deviate to avoid encountering the obstacle. The output layer has a single neuron. The Levenberg Marquardt (LM) algorithm is used for weight adjustment in the Network, which is a function optimization technique and is reserved to offline because of the difficulties to implement a true iterative function. The activation function is the Purelin function, which is the output calculated from the net input and is linear in nature.

Keeping the training speed under consideration, only one hidden layer, consisting of eight neurons is used. The activation function applied is the Tan sigmoid function, whose output values ranges from -1 to +1. With Feed-forward network approach, the signals are directed forward and the error is transported backwards. The MSE (Mean Square Error) is calculated to improve the efficiency of the controller.

V. EXPERIMENTAL DATA AND ANALYSIS

The Neural Network controller used fetches the satisfactory results. In Fig.6, the plot shows the desired output pattern and the results obtained from the training. The ANN provides results close to the desired values. It can be inferred that the ANN results are more accurate and precise. The green coloured plot shows the NN output tracing the training pattern. The small variations indicate the weight changes that occur as the epochs get incremented.

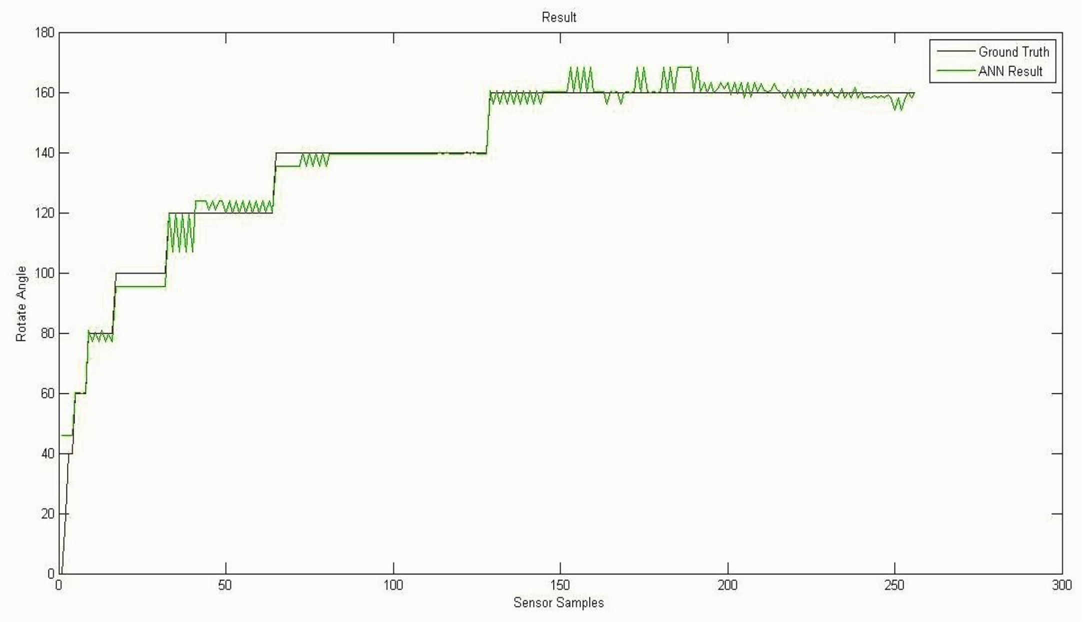

Fig. 6: ANN results versus desired results

The data in table 1 shows the patterns provided for training and the trained pattern from the NN controller for comparison. By comparing the output patterns, one may infer that, the trained output has higher accuracy and gives precise value up to fourth decimal point. The NN performed thousand iterations. The best training performance is observed at 190.3993 at epoch 886.

TABLE I: Samples of training and trained data

No.	Input to the Neural Network (Output from the sensors)								Output (Rotation angle)	Output (Rotation angle)
	S1	S2	S3	S4	S5	S6	S7	S8	Training data	Trained data
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	20	22.5689
3	0	0	0	0	0	0	1	0	40	34.3658

4	0	0	0	0	0	0	1	1	40	45.8568
5	0	0	0	0	0	1	0	0	60	60.0000
6	0	0	0	0	0	1	0	1	60	60.0545
7	0	0	0	0	0	1	1	0	60	59.9999
8	0	0	0	0	0	1	1	1	60	60.0545
9	0	0	0	0	1	0	0	0	80	81.0079
10	0	0	0	0	1	0	0	1	80	76.9874

The Mean Square Error (MSE) obtained throughout the training process is depicted in Fig.7. From the plot, it can be inferred that as the training advances and synaptic weights get modified the MSE considerably reduces. The LM algorithm optimises the error to minimum. The MSE at the end of training is $7.0809e^{-6}$. The results prove the effectiveness of the applied algorithm in reducing the MSE considerably throughout the training process.

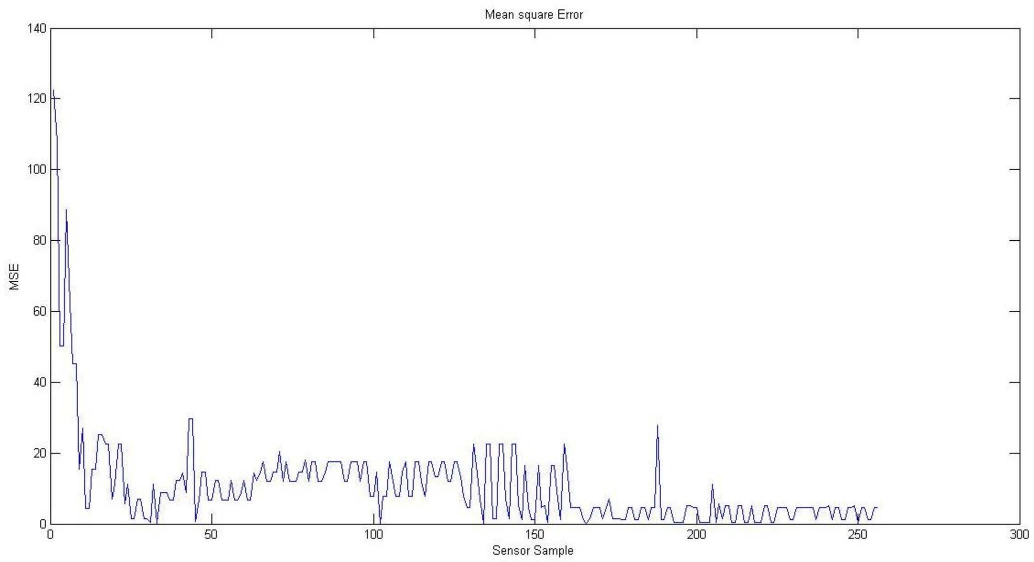

Fig. 7: Mean Square Error plot

The mobile robot system that is developed for the evaluation of the practical effectiveness of the proposed approach is shown in Fig.8. The user controlled the direction of the robot using the GUI. In the presence of an obstacle, the path is seen to be avoided and the navigation control is hence achieved along with the speed control.

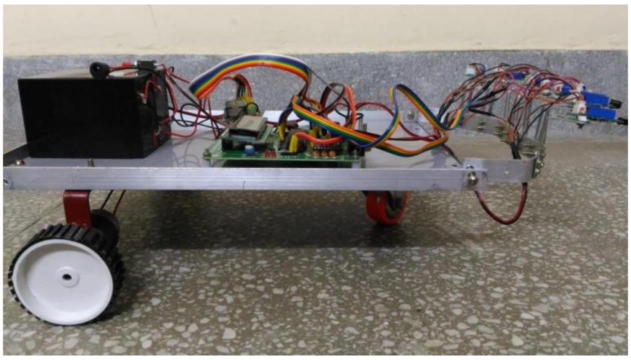

Fig. 8: Complete hardware setup

VI. FUTURE SCOPE AND CONCLUSION

The neural network training produced satisfactory results. The integration of the NN controller to the mobile robot system built, showed satisfactory performance .The navigation control has achieved well. The PI control proved to be a better control strategy in speed control of the robot. As a result of the delay in the serial communication with the MATLAB and controller, the performance witnessed a delay. This can be avoided with better communication methods, the delay may be avoided.

REFERENCES

[1] G. N. Tripathi 1 and V.Rihani, " Motion planning of an autonomous mobile robot using artificial neural network", CS & IT- CSCP 2012

[2] Timothy A. Zimmerman, "Neural Network Based Obstacle Avoidance Using Simulated Sensor Data", ASEE 2014 Zone I Conference, April 3-5, 2014.

[3] A. Medina-Santiago, J. L. Camas-Anzueto, , J. A. Vazquez-Feijoo, H. R. Hernández-de León, Mota-Grajales, "Neural Control system in Obstacle Avoidance in Mobile Robots Using Ultrasonic Sensors", Journal of Applied Research and Technology, Vol. 12, February 2014.

[4] Kai-Hui Chi and Min-Fan Ricky Lee, "Obstacle Avoidance in Mobile Robot using Neural Network", IEEE, 2011.

[5] Sanjay Singh and Dr. A. K. Pandey, Dipraj, "Design of Pi Controller to Minimize the Speed Error of D.C. Servo Motor", International Journal of Scientific & Technology Research Volume 1, Issue 10, November 2012

[6] V Mustafa I. Hamzah and Turki Y. Abdalla, "Mobile Robot Navigation using Wavelet Network and Fuzzy Logic," International Journal of Computer Applications (0975 – 8887), Volume 79 – No10, October 2013

[7] Carlos Gershenson, "Artificial Neural Networks for Beginners".

[8] V Rumelhart, D. and J. McClelland, "Parallel Distributed Processing", MIT Press, Cambridge, Mass, 1986.