International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 3, Issue 6, November-December 2014 **ISSN 2278-6856**

Development of MBP for AVR Based Controlled Autonomous Vehicle

Ms.D.S.Nikam[1], Dr.S.T.Gandhe[2] , Ms.J.N.Phasale[3]

[1]Sandip institute of technology and research center ,Nashik university of Pune ,India

[2] Sandip institute of technology and research center ,Nashik university of Pune ,India

[3]SGDCOE,Jalgaon,NMU Univercity,India

Abstract

Nowadays application of autonomous vehicle increases .An Autonomous vehicle is an artificially intelligent vehicle capable of traveling in unknown and unstructured environments independently. This paper proposes, design of a low cost autonomous vehicle based on backpropagation algorithm of neural network controlled by AVR for navigation in unknown environments. The neural network running inside the AVR is a multilayer feed-forward network. The advantage of using neural approach over the conventional inverse kinematic algorithms is that ANNs can avoid time consuming calculations. Furthermore, in a manner typical of ANNs, it would be very easy to modify the learned associations upon changes in the structure of robot manipulators. With reference to the control paradigm, ANNs have the ability to approximate arbitrary nonlinear functions which is an essential requirement in the design of controllers for nonlinear dynamic systems. Because of their learning and adaptive features, ANNs can be trained to adaptively control various nonlinear systems. For training neural network MBP is used. Multiple Back propagation is a free software application for training neural network with back propagation & multiple back propagation algorithm. It is easy to use and most important feature is that generate C code for the trained network. This code is compatible with ardiuno uno so it easily burned onto ATMEGA 328. In this paper we describe thoroughly software and hardware part and conclude the result from simulation graph. The hardware of vehicle is equipped with three ultrasonic sensors for hurdle distance measurement, all interfaced to a low cost ATMEGA 328 microcontroller. The microcontroller processes the information acquired from the sensors and generates robot motion commands accordingly through neural network.

Keywords:- Autonomous vehicle, Multilayer Feed forward neural network, Back propagation algorithm, MBP, ultrasonic sensor, At Mega 328

1. INTRODUCTION

In recent years, ANNs have been successfully used in various applications such as system identification and control, robotics, pattern recognition and vision. One important application of ANNs is in the area of robotics. In particular, ANNs have been used to compute inverse. Particularly in autonomous vehicle [2]. The key requirement of autonomous vehicle is the navigation. Navigation is the ability of the mobile robot to reach their target safely without human assistance. The evolvement of soft-computing paradigms has provided a powerful tool to deal with mobile robot navigation process. Among all the soft-computing methods fuzzy logic based decision-making and neural networks have been found to be the most attractive techniques that can be utilized for this purpose. Fuzzy system is tolerant to noise and error in the information coming from the sensory system, and most importantly it is a factual reflection of the behavior of human expertise. In general, there are two approaches to the application of fuzzy logic in mobile robot navigation, namely, behavior-based approach [5-9] and classical fuzzy rule-based approach. However, the design of fuzzy logic rules is often reliant on heuristic experience and it lacks systematic methodology, therefore these rules might not be correct and consistent, do not possess a complete domain knowledge, and/or could have a proportion of redundant rules. Furthermore, these fuzzy logic rules cannot be adjusted or tuned on real-time operation, and the off-line adjustment of their parameters is a time consuming process. Another problem could be raised when better precision is needed which is the huge expansion in the fuzzy rule-based system [7]. On the other hand, several successful reactive navigation approaches based on neural networks have been suggested in the literature . In spite of the different suggested network topologies and learning methods. However, neural networks that can be implemented with relatively modest computer hardware could be very effective and useful. This paper proposes a low cost implementation of a autonomous vehicle based on artificial neural network (ANN). The propose neural network is multilayer feed-forward network with back propagation learning algorithm consisting of three layers: an input, hidden and output layer. The input layer is made out of three neurons that are fed by the ultrasonic sensors which are used to detect obstacle in the path of the vehicle. The hidden layer is made out of six neurons and its purpose is to read input data and multiply them by a certain weight and then forward the results to the next layer. The output layer is made out of one neuron that are directly linked to the motors which control its movement and its direction. The proposed ANN uses a mix of activation functions including Sigmoid for the hidden

neurons and linear for the output neurons. Moreover, the model employs a supervised learning approach using the back-propagation algorithm to train the network in offline mode [5].An autonomous vehicle is realized using a single readily available ATMEGA 328 microcontroller. It is 8-bit Atmel Microcontroller with 328Kbytes In- System Programmable Flash memory. The network is trained offline by using MBP with tangent-sigmoid being the firing function of neurons. Because linear functions require much lesser computation power than nonlinear functions, tangent-sigmoid function is approximated as piecewise linear function for implementation in microcontroller. This helps in reducing computational time. MBP is an application for training the neural network with back propagation algorithm, which can be veiwed as generalization of Back propagation algorithm.

2. METHODOLOGY

2.1 Basics of neural network

Neural networks are composed of simple elements operating in parallel. They are connected each other via link and each link is associated with numeric value called as a weight. All the weights in a network are generally initialized to a random value between zero and one. Learning in a neural network typically performs by adjusting a weight. These elements are inspired by biological nervous systems. Fig. shows simple neural network.

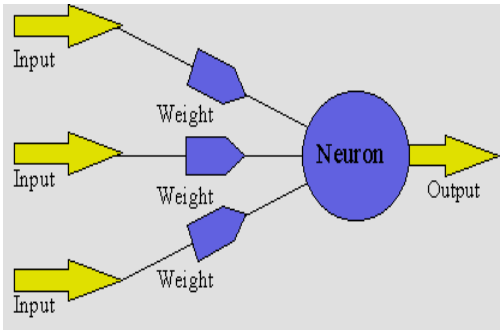

Figure 1 Simple Neural network

There are two types of learning techniques used by neural network supervised and unsupervised. In supervised learning inputs and targets are known. This learning progression allows the network to identify many patterns and further generalize to new and unseen patterns. Such type of training is called supervised learning which uses classified pattern information to train the network in offline mode. On the other hand, there exists what so called the unsupervised learning which uses only minimum information without pre-classification to train the network while being in online mode [2]. Some of the most successful supervised learning approaches are feed-forward and back-propagation; while, the most successful unsupervised learning approaches are the Hebbian and the competitive learning rule.Here we are using *Back* Propagation Algorithm.

2.2. Back propagation algorithm

A Back Propagation network learns by example. You give the algorithm examples of what you want the network to do and it changes the network's weights so that, when training is finished, it will give you the required output for a particular input. Back Propagation networks are ideal for simple Pattern Recognition and Mapping Tasks4.As just mentioned, to train the network you need to give it examples of what you want– the output you want (called the Target) for a particular input as shown in Figure. So, if we put in the first pattern to the network, we would like the output to be 0 1 as shown in figure 3.2 (a black pixel is represented by 1 and a white by 0 as in the previous examples). The input and its corresponding target are called a Training Pair. Once the network is trained, it will provide the desired output for any of the input patterns. Let's now look at how the training works. The network is first initialized by setting up all its weights to be small random numbers – say between –1 and +1. Next, the input pattern is applied and the output calculated (this is called the forward pass). The calculation gives an output which is completely different to what you want (the Target), since all the weights are random. We then calculate the Error of each neuron, which is essentially: Target - Actual Output (i.e. What you want ,What you actually get). This error is then used mathematically to change the weights in such a way that the error will get smaller. In other words, the Output of each neuron will get closer to its Target (this part is called the reverse pass). The process is repeated again and again until the error is minimal.

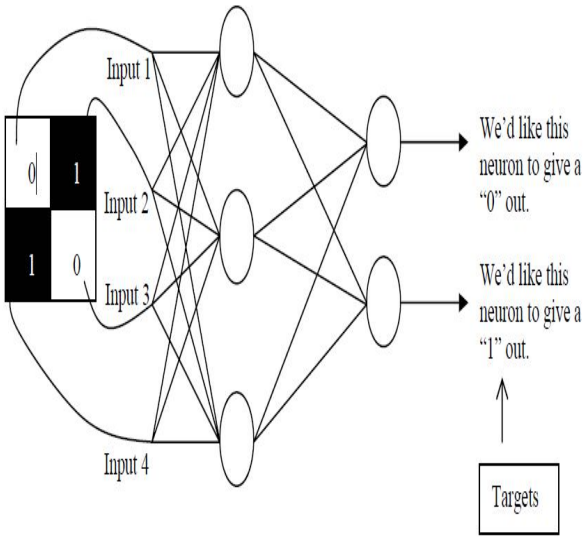

Figure 2 Back Propagation Network

3 PROPOSED SYSTEM

3.1 Block diagram and flowchart

In proposed system we are using ATMEGA 328A controller with ultrasonic module. Three ultrasonic sensor mounted on vehicle which continuously measure the distance from obstacle. Neural network take decision with the help of controller. Block diagram of the system is shown in fig. 3 and Flowchart of the project is shown in fig.4

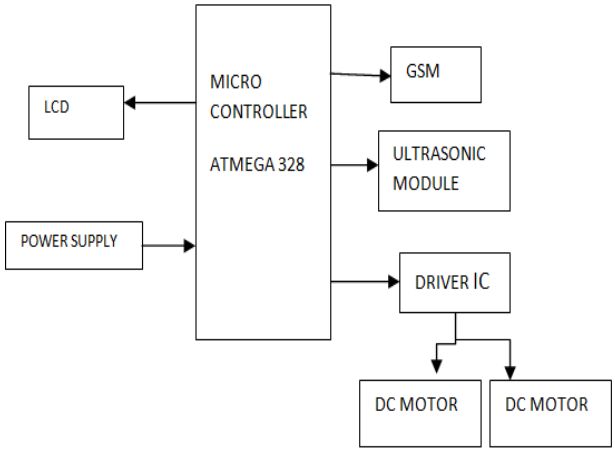

Figure 3 Block Diagram

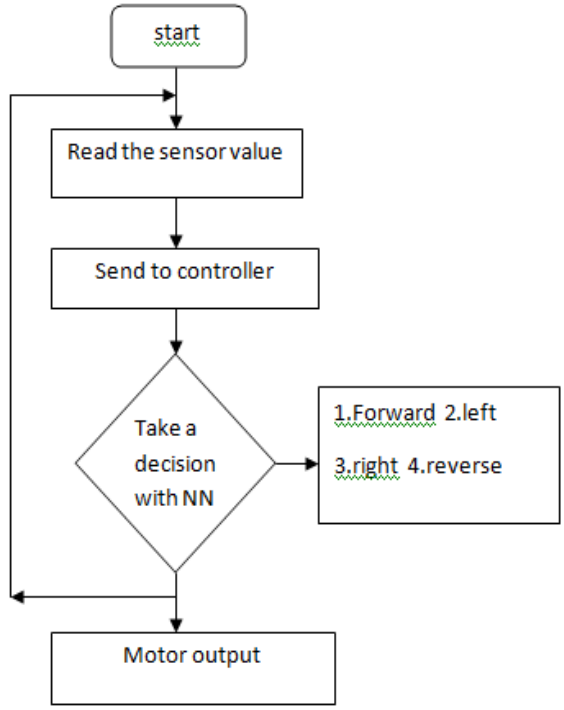

Figure 4 flowchart of proposed system

3.2 Implementation

For the proposed system we use multi layer feed-forward network with back propagation learning algorithm. The employed configuration contains 3 neurons in the input layer, 6 in the hidden layer and one in the output layer as shown in Fig. 3. Three ultrasonic sensor mounted on vehicle gives the distance from obstacle, these distance shown on lcd .as shown in table motor output is decided on the basis of the sensor information. for simplicity here we are considering 4 range for the input sensor very far denoted as 0(>24inch),far denoted as 1(16-23inch),near denoted as 3(10-22inch),very near denoted as 4 (<10 inch).similarly left and right sensor is used to detect presence of obstacle denoted by 1(<16 inch)and absence of obstacle denoted as 0 (>16inch).Depending upon these information output direction 1for front, 2 for left,3 for right and 4 for reverse is decided with the help of rule table. multiple back propagation (MBP) free coder is used for training the neural network and generate the c code for it.

Table 1: Rules for training examples

Range sensor	Left	Right	Motor Output
0	0	0	1
1	0	0	1
2	0	0	2
3	0	0	3
0	0	1	1
1	0	1	1
2	0	1	2
3	0	1	2
0	1	0	1
1	1	0	1
2	1	0	3
3	1	0	3
0	1	1	1
1	1	1	4
2	1	1	4
3	1	1	4

As shown in table 1 rules for training the networks are written into separate excel file and saved the file with .Csv extension, all the things written in csv file must be in numerical form there is no need to give the explanation of input and output .The range sensor input conditions replace by 0,1,2,3 respectively showing vfar, far, near, vnear distance information. For training the networks specifying the topology as a 3-6-1 Where 3 input neuron, 6 hidden neuron and 1 output neuron. Output neuron describe as integer value for 4 output condition i.e front, left, right and reverse represented by 1,2,3,4 respectively. Then .csv file uploaded into train bar. Once it loaded then specifies the activation function. For this project we are using tan sigmoid function for hidden neuron and liner function for output neurons. Then randomize the weights between -1 and 1.Now network is ready to train so by just clicking on train button network start training on screen MSE graph occurred for space and main network. Network weights of each layer and each neurons are adjusted automatically to minimize the error and final adjusted weights for each layer are shown in table 2,3,and 4 respectively. We also see the graph of output vs desired training signal as shown in fig.5& me 6 respectively. Finally generated C code burned onto the Atmega 328 controller. We were taking the trial of the vehicle in indoor environment with 4 stationary obstacle as shown in fig. Every time it displays the distance from each side and then it moves in direction where obstacle distance is high. Therefor it avoiding the heating of obstacle.

4. EXPERIMENTAL RESULTS

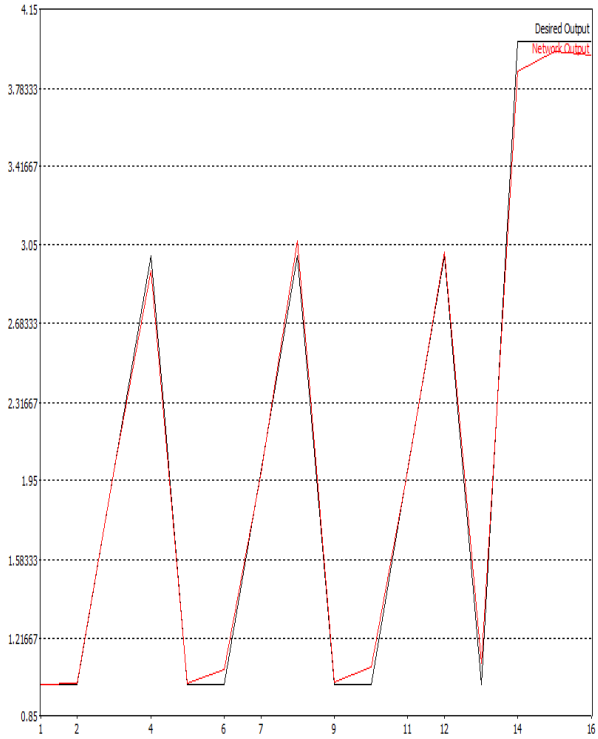

Figure 5 Desired output vs network output

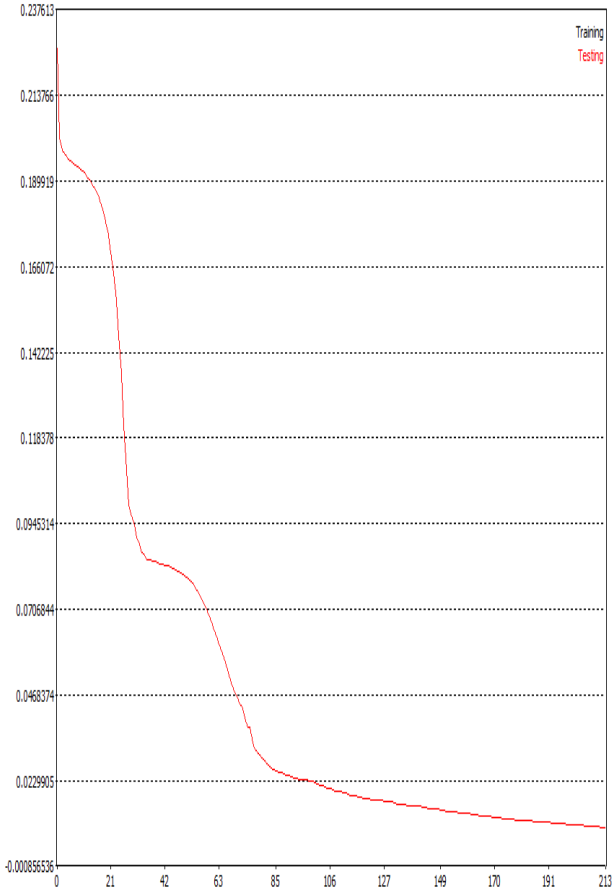

Figure 6 RMS graph

Table 2: weights for inputs layer to the hidden layer1

	from the input layer			
to the 1th hidden layer	bias	1th neuron	2th neuron	3th neuron
1th neuron	-1.73838	0.713217	1.3676	1.6746
2th neuron	-1.63972	-4.3637	-0.48314	-0.570873
3th neuron	1.04543	2.18745	-1.32859	1.78689
4th neuron	-1.1267	-2.35557	-2.21224	1.80617
5th neuron	0.308976	0.379467	-0.989932	0.849218
6th neuron	-1.48347	-0.0571192	1.58705	1.97566

Table 3: weights for 1st hidden layer to the 2nd hidden layer

	from the 1th hidden layer						
to the 2th hidden layer	bias	1th neuron	2th neuron	3th neuron	4th neuron	5th neuron	6th neuron
1th neuron	-0.703471	1.60058	-0.825934	0.112004	0.163796	-0.658783	1.23028
2th neuron	-0.0964405	1.09384	-1.06592	0.563866	-0.625206	-0.548991	0.800301
3th neuron	0.638868	-1.90653	2.32366	-1.18608	1.80812	-1.13545	-0.939449
4th neuron	-0.675393	-0.950789	1.72286	-0.962069	1.82552	0.425677	-0.883655
5th neuron	-0.339347	0.308036	1.62649	-1.07214	1.13003	0.0969803	-1.60175
6th neuron	-0.16447	-0.497883	2.1682	-1.4644	0.928439	-0.457049	-0.801422

Table 4: weights for 2nd hidden layer to the output layer

	from the 2th hidden layer						
to the output layer	bias	1th neuron	2th neuron	3th neuron	4th neuron	5th neuron	6th neuron
1th neuron	1.42476	2.02601	2.00159	-3.81984	-2.60545	-2.06906	-2.12128

Figure 7 Navigate in front direction

Figure 8 Navigate without heating obstacle

5.CONCLUSION

In this paper, design of a low cost autonomous vehicle based on neural network is presented for transportation of light weight equipment inside the college campus. Here we considered MBP software to train the neural network . From the simulation we conclude that in RMS graph the mean squared errors are decreasing in other word we said that network is learning. Mean squared error is the average square difference between actual value and target value. Here the result are reasonable because of final mean square error is small and in fig.4 network output follows the desired output appropriately. Weights and biases for input, output and hidden layers are seen after training of the network and shown in table 2,3,4 respectively. By clicking on generate c code we get embedded c code. The same code burn onto the At mega 328 Controller and necessary movements of the vehicle is done accordingly. The vehicle is tested in various indoor environments containing the obstacles and is found to avoid the obstacles for most of the times. Experimental results have shown the validity of the system which avoids the obstacles and perform the navigation of the vehicle successfully.

REFERENCES

[1] Umar Farooq Muhammad Amar, Eitzaz ul aq,Muhammad Usman Asad,Hafiz Muhammad Atiq"Microcontroller based Neural NetworkControlledLow Cost Autonomous Vehicle."IEEE computer society.pp. 96-100 ,2010

[2] Youssef Bassil" Neural Network Model for Path-Planning Of Robotic Rover Systems ". International Journal of Science and Technology (IJST), E-ISSN: 2224-3577, Vol. 2, No. 2, February, 2012

[3] Schraft R.D., "Mechatronics and Robotics for Service Applications", IEEE Robotics &Automation Magazine, pp. 31-37, December 1994.

[4] Dipti Srinivasan, , Min Chee Choy, and Ruey Long Cheu,"Neural Networks for Real-Time Traffic Signal Control"IEEE transactions on intelligent transportation systems, vol. 7,pp. 261-272 , no. 3, September 2006

[5] Harris G., "Robotics Decontamination Keeps Operators Clear of Danger", Industrial Robot, 20:3, pp. 30-33, 1993.

[6] J. F. Martins, V. Fernão Pires, , and A. J. Pires, "Unsupervised Neural-Network-Based Algorithm for an On-Line Diagnosis of Three-Phase Induction Motor Stator Fault" ,IEEE industrial electronics , VOL. 54, NO. 1,pp 259-264, February 2007

[7] Khaldoun K. Tahboub, Munaf S. N. Al-Din, "A Neuro-Fuzzy reasoning system for mobile robot navigation," Jordan Journal of Mechanical and Industrial Engineering, vol. 3 (1), pp. 77-88, March 2009

[8] Y. Zhou, D. Wilkins, R. P. Cook, "Neural network control for a fire-fighting robot," University of Mississippi.

[9] Weisbin C.R., D.Lavery, "NASA Rover and Telerobotics Technology Program" , IEEE Robotic & Automation Magazine, pp. 14-20, December 1994.

[10] Evans J., Krishnamurthy B., "Handling Real World Motion Planning A Hospital Transport Robot", IEEE Control Systems, vol.12 pp.15-19,February 1992

AUTHER

D.S.Nikam was born in Nashik, India. completed B.E in E&TC Engineering from Cummins college of engineering, Pune In 2010 and pursuing M.E. in E&TC from Sandip institute of technology and research center, Nashik. Presently work as a Assistant professor in SIER,Agaskhind,Nashik

Dr.S.T.Gandhe he has received the .D from VNNIT, Nagpur, M.S. degrees in Electrical, electronics and communication Engineering from university Baroda in 1994-1996. Presently working as a principal in Sandip institute of technology and research center, Nashik. His area of research interest in DSP Artificial neural network, Fuzzy logic.

J.N.Phasale was born in Nashik, India. completed B.E completed B.E. in E&TC Engineering from Cummins coll Sanjivani college of engineering, Kopergaon In2010 and pursuing M.E. in E&TC from SGDCOE, Jalgaon.