

# Design and Implementation of Neural Network Based Controller for Mobile Robot Navigation in Unknown Environments

Umar Farooq, Muhammad Amar, Muhammad Usman Asad, Athar Hanif, and Syed Omar Saleh

**Abstract**—In this paper, design of an intelligent autonomous vehicle is presented that can navigate in noisy and unknown environments without hitting the obstacles in its way. The vehicle is made intelligent with the help of two multilayer feed forward neural network controllers namely ‘Hurdle Avoidance Controller’ and ‘Goal Reaching Controller’ with back error propagation as training algorithm. Hurdle avoidance controller ensures collision free motion of mobile robot while goal reaching controller helps the mobile robot in reaching the destination. Both these controllers are trained offline with the data obtained during experimental run of the robot and implemented with low cost AT89C52 microcontrollers. The computational burden on microcontrollers is reduced by using piecewise linearly approximated version of tangent-sigmoid activation function of neurons. The vehicle with the proposed controllers is tested in outdoor complex environments and is found to reach the set targets successfully.

**Index Terms**—Navigation in complex environments, neural network, hurdle avoidance behavior, goal reaching behavior, real time implementation.

## I. INTRODUCTION

Navigation is the ability of a mobile robot to reach the set targets by avoiding obstacles in its way. Thus essential behaviors for robot navigation are obstacle avoidance and goal reaching [1], [2]. Conventional control techniques can be used to build controllers for these behaviors; however, the environment uncertainty imposes a serious problem in developing the complete mathematical model of the system resulting in limited usability of these controllers. Thus some kind of intelligent controllers are required that can cope with the changing environment conditions. Amongst the various artificial intelligence techniques available in literature, neural networks offer promising solution to robot navigation problem because of their ability to learn complex non linear relationships between input sensor values and output control variables. This ability of neural networks has attracted many researchers across the globe in developing neural network based controllers for reactive navigation of mobile robots in indoor as well as outdoor environments. In [3], a collision free path between source and destination is constructed based on

neural networks for mobile robot navigation in partially structured environments. The proposed scheme uses two neural networks to accomplish the task. First neural network is a principal component analysis (PCA) network with generalized Hebbin rule and is used to find a free space using ultrasonic range finder data. The second neural network is a multilayer perceptron (MLP) network with back-propagation training algorithm and is used to find a safe direction for robot movement while avoiding the nearest obstacles. The proposed scheme is implemented in real time on Intel Pentium 350 MHz processor and robot is found to avoid all the obstacles in reaching the destination from start point. In [4], kohonen and region-feature neural networks have been used to address global self localization problem of mobile robot which is an essential behavior to determine the current position of the robot during navigation. The robot with these controllers learns the regions of space just like optical character recognition with the help of sensory data gathered from exploring the environment. Experimental results have shown that the proposed technique is robust owing to time-, translational-, and rotation invariant. In [5], mobile robot navigation problem is solved with the help of local model networks. This network is a set of sub-models that represent the dynamic system be modeled at various operating points. Each sub-model is a feed forward neural network trained with back-propagation algorithm. The output of these sub-models is weighted with the help of a radial basis function neural network to generate motion commands for robot. The performance of local model network is compared with both multilayer perceptron and radial basis function networks with time taken by the robot to reach the destination as performance index and is found to outperform both these networks. In [6], design of a navigation controller composed of three neural sub-networks is presented. The first two controllers are responsible for most important behaviors of intelligent vehicle namely target localization and obstacle avoidance. Both these controllers are classifiers and are trained with standard supervised back propagation techniques. The target localization controller maps the temperature fields around the robot to the angular sector in which the target lies while obstacle avoidance controller maps the sensor values to thirty local obstacle configurations. The third neural network acts as supervisor and is responsible for the final decision based on the outputs of first two neural controllers. This controller is trained by a variant of the associative reward-penalty algorithm for learning. Due to this hierarchical structure, complexity of system has been reduced resulting in faster response time. Our work is similar to that

Manuscript received May 27, 2013; revised August 22, 2013.

Umar Farooq, Muhammad Amar, and Syed Omar Saleh are with Department of Electrical Engineering, University of The Punjab Lahore (e-mail: engr.umarfarooq@yahoo.com; amar.ete611@yahoo.com; omar\_saleh81@yahoo.com).

Muhammad Usman Asad and Athar Hanif are with Department of Electrical Engineering, The University of Lahore (e-mail: usmansad01@hotmail.com; athar.hanif@ee.uol.edu.pk).

reported in [6]. However, instead of using the third neural network, we have used simple decision logic to generate the final motion commands for the robot.

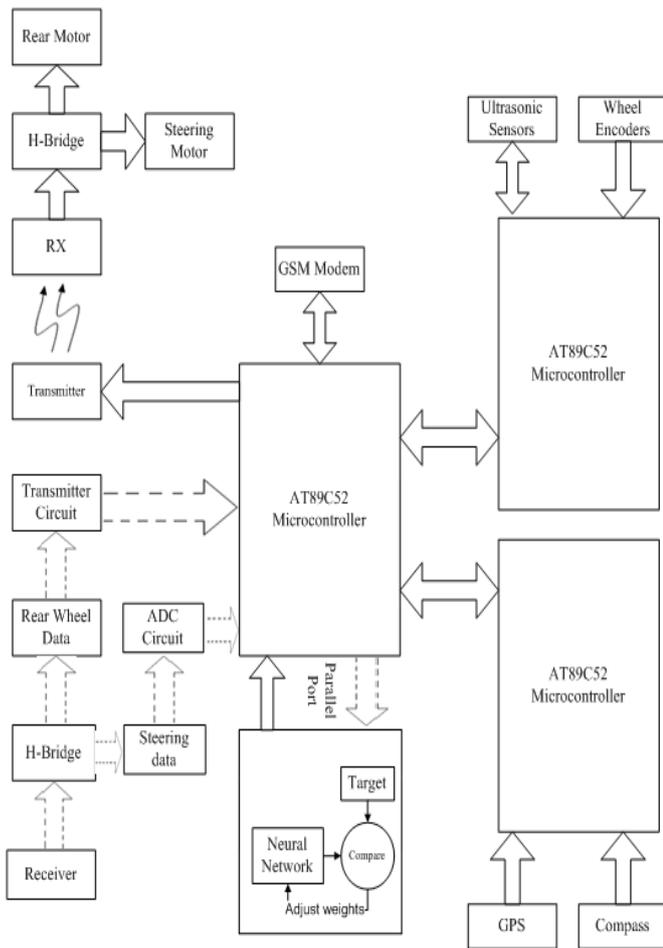


Fig. 1. System block diagram

The present work describes the design of an autonomous vehicle that uses two neural network controllers for navigation in outdoor environments. Both these controllers are feed forward neural networks trained with back-propagation algorithm and are named as ‘Hurdle Avoidance Controller’ and ‘Goal Reaching Controller’. The task of hurdle avoidance controller is to ensure collision free motion of the vehicle amongst obstacles. It accepts input from two ultrasonic sensors mounted in front of the vehicle in the form of distance to obstacles and generates commands for steering and rear motors to avoid obstacles. The task of goal reaching controller is to move the vehicle from source location to destination location. It accepts inputs from GPS receiver and digital compass in the form of distance to goal location and heading error between vehicle and goal orientation respectively and generates steering angle commands to keep the vehicle aligned with the destination. The outputs of both these controllers is fed decision logic controller which output the final motion commands for the robot. The proposed controllers are trained offline in MATLAB® environment with the data obtained during experimental run of the robot and implemented in real time using readily available AT89C52 microcontrollers. The system block diagram is shown in Fig. 1 where off line training is shown by the dashed lines.

## II. SYSTEM ARCHITECTURE

### A. Vehicle Chassis

A four wheeled car type vehicle robot is selected for experimentation which is a modified version of readily available RC car.

### B. Steering Circuit

A potentiometer is connected to the steering of vehicle for obtaining steering angle information in order to train the neural network. This steering information is converted to digital form with the help of ADC0808 analog to digital converter and is fuzzified to define nine regions: extreme left, large left, medium left, small left, straight, small right, medium right, large right and extreme right. The rear wheel information is classified as either forward or backward. The fabricated steering circuit is shown in Fig. 2.

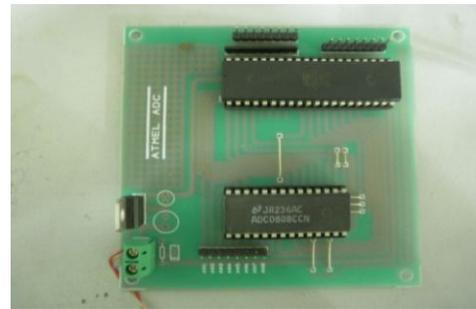


Fig. 2. Steering circuit

### C. Motor Drive Circuit

To drive the dc motors from microcontroller, a motor interfacing board is designed using opto-couplers 4N25 and motor driver ICs L298N. Opto-coupler is used to provide isolation between microcontroller and motor drive units while L298N drives the motor. IC L298N contains two H-bridges each capable of handling currents up to 2A. These bridges are connected in parallel to enhance the current rating. The motor interface board is shown in Fig. 3.



Fig. 3. Motor drive circuit

### D. Ultrasonic Sensors

SRF05 ultrasonic sensors are used for acquiring distance information. A short 10uS pulse is applied to the trigger input to start the ranging from controller. The SRF05 sends out an 8 cycle burst of ultrasound at 40 kHz and raise its echo line high (or trigger line). It then listens for an echo, and as soon as it detects one it lowers the echo line again. The echo line is therefore a pulse whose width is proportional to the distance to the object. By timing the pulse the range of a nearby object

is calculated. Two such sensors are mounted on front side of the robot. The outputs from these sensors are fuzzified to define five regions: very near, near, medium, far, very far and is represented by three bits for each sensor as shown in Fig. 4.

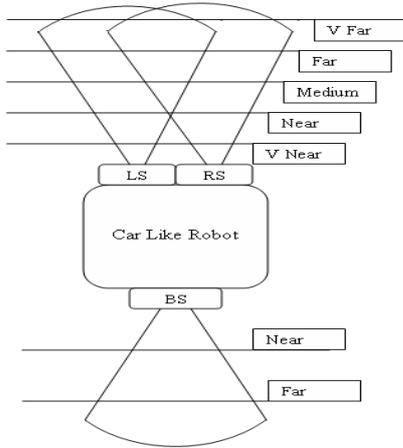


Fig. 4. Sensor regions

### E. Digital Compass

Digital compass is built by modifying readily available needle compass. A total of 18 LDRs are mounted in a circle fashion around the needle compass. These LDRs are shined from above with the help of LEDs. Whenever the direction of motion is changed, a particular LDR or pair of LDRs is blocked from shining by the needle. This information is converted into 0-5V range with the help of LM311 operational amplifiers which are connected as comparators. In this way, 100 degree resolution is obtained. The fabricated compass is shown in Fig. 5.

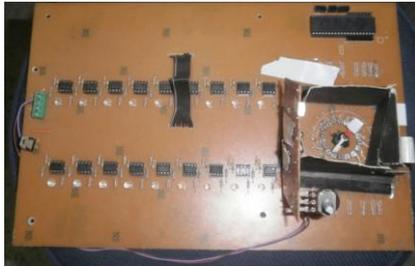


Fig. 5. Digital compass

### F. Wheel Encoders

Slotted disk having 50 slots with U-shape sensor comprise the wheel encoder for measuring the distance traveled by the robot. The sensor provides 50 pulses to microcontroller for one complete revolution of the wheel.

### G. GSM Modem

An SIM300D GSM modem is used to change the destination place on run time. AT commands are used by the microcontroller to communicate with the GSM modem. The modem also informs the central station about the track history of the vehicle and any emergency situation occurred.

### H. GPS Receiver

An M89 GPS receiver is used to get position information of robot in the form of latitude and longitude values (N, E) as it

moves in out door environment. Since the input to the goal reaching controller will be the distance information between robot current position and goal location, haversine formula [7] is employed to determine this distance, described as:

$$\begin{aligned} \Delta lat &= lat_2 - lat_1 \\ \Delta long &= long_2 - long_1 \\ a &= \sin^2(\Delta lat/2) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2(\Delta long/2) \quad (1) \\ c &= 2 \cdot a \cdot \tan^{-1}(\sqrt{a}, \sqrt{1-a}) \\ \Delta R &= R_e \cdot c \end{aligned}$$

where,

$R_e$  = Earth's radius (mean radius = 6,371km)

$\Delta R$  = Distance between robot current position

$(lat_1, long_1)$  and goal position  $(lat_2, long_2)$

$\Delta R$  is fuzzified to define five regions: very near, near, medium, far and very far. The experimental robotic vehicle equipped with aforementioned sensors is shown in Fig. 6.



Fig. 6. Experimental Robotic Vehicle

## III. NEURAL CONTROLLER DESIGN

The mobile robot navigation in outdoor environments is achieved with the help of two neural controllers namely hurdle avoidance and goal reaching. Both these controllers use two layer feed-forward networks with back propagation learning algorithm and are designed using MATLAB® programming environment [8]. The employed configuration for hurdle avoidance controller contains 5 neurons in the hidden layer and 2 in the output layer, as shown in Fig. 7, while goal reaching controller uses 4 neurons in the hidden layer and 2 neurons in the output layer as shown in Fig. 8. The numbers of neurons in hidden layer are selected on trial and error basis and kept at minimum for reducing the complexity. The distance to hurdle information from three ultrasonic sensors ( $LS/RS/BS$ ) is provided as inputs to the hurdle avoidance controller which generates control commands for steering ( $SA$ ) and rear motors ( $F/B$ ) while distance between robot current position and goal location ( $\Delta R$ ) along with the heading error between robot and goal orientation ( $\Delta \theta$ ) are fed as inputs to the goal reaching controller which generates commands for steering motor and gives information whether destination has been reached or not ( $DB$ ). These inputs/outputs are scaled as:

$$\begin{aligned} 0 &\leq LS, RS \leq 5 \\ 0 &\leq \Delta R \leq 5 \\ -4 &\leq \Delta \theta \leq 4 \\ -4 &\leq SA \leq 4 \end{aligned} \quad (2)$$

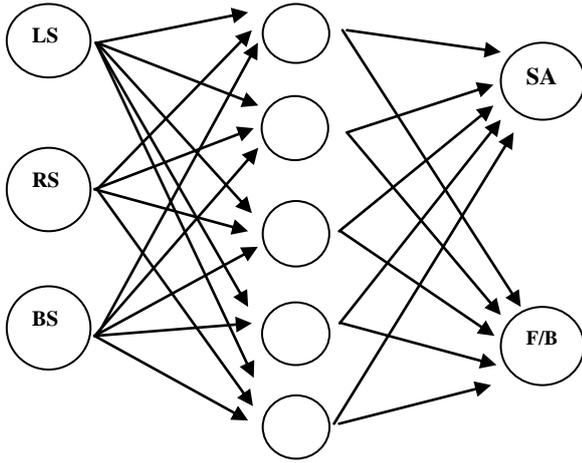


Fig. 7. Hurdle avoidance neural network controller

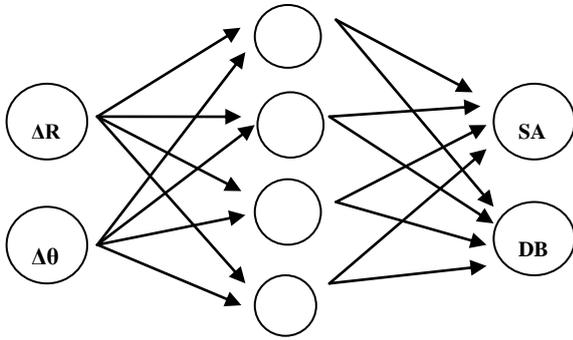


Fig. 8. Goal reaching neural network controller

The activation function used for hidden layer in both neural controllers is tangent-sigmoid function while pure linear function is employed in output layer. The data used for training the neural networks is gathered by driving the vehicle with the help of remote control in complex environments. An exemplary training data for hurdle avoidance and goal reaching controllers is shown in Table I and II respectively. This data is divided into two sets: training data set and validation data set. The neural networks with the training data sets are trained offline in MATLAB® environment. During training, for each sample value, error is calculated between the desired output and network calculated output. The error is then minimized by using back propagation training algorithm. The algorithm minimizes the error by updating the weights and biases of the network. The formula for updating  $w_{ij}$ , the weight of the link between input unit  $i$  and output unit  $j$ , at time  $t+1$  is:

$$W_{ij}(t+1) = W_{ij}(t) + \eta[t_j(t) - o_j(t)]i_i(t) + \alpha\Delta W_{ij}(t-1) \quad (3)$$

Where,  $\eta$  is the learning rate (defined as 0.3),  $t_j(t)$  and  $o_j(t)$  are the target output and actual output from unit  $j$  respectively at time  $t$ ,  $i_i(t)$  is the input at unit  $i$  at time  $t$ ,  $\alpha$  is the learning momentum (also defined as 0.3) used for convergence of network output to desired behavior by speeding up the iterative process, and  $\Delta w_{ij}(t-1)$  is the weight update on the link from unit  $i$  to unit  $j$  in the previous iteration. After performance goal is met in training phase, the networks are

tested with validation data set. This data set is used to avoid over-fitting the network to the training data. The training error graph showing the performance of hurdle avoidance network is shown in Fig. 9 while for goal reaching network, it is shown in Fig. 10.

TABLE I: EXEMPLARY TRAINING DATA FOR HURDLE AVOIDANCE CONTROLLER

Function	LS	RS	BS	SA	F/B
If LS measures very far and RS also measures very far then car will go forward at high speed	5	5	1/0	0	4
If LS measures far and RS measures very far then car will turn at small rate towards right and go forward at high speed	4	5	1/0	1	3
If LS measures far and RS measures medium then car will turn at medium rate towards left and go forward at medium speed	4	3	1/0	-2	2
If LS measures medium and RS measures very near then car will turn at large rate towards left and go forward at slow speed	3	1	1/0	-3	1
If LS measures very near and RS measures very near and BS measures far then car will turn at extremely high rate towards left while reversing	1	1	1	-4	-4
If LS, RS and BS measures very near then car will stop, turn on its horn and wait for the sensor values to change	1	1	0	0	0

TABLE II: EXEMPLARY TRAINING DATA FOR GOAL REACHING CONTROLLER

Function	$\Delta R$	$\Delta\theta$	SA	DB
If destination is at very far distance and the current heading angle is on the extreme left side of destination angle then turn at a small rate towards right to align with the goal	5	-4	1	0
If destination is at medium distance and current heading angle is on the small left side of the destination then turn at small rate towards right to align with the goal	3	-1	1	0
If destination is at medium distance and current heading angle is on the extreme left side of the destination then turn at medium pace towards right to align with the goal	3	-4	2	0
If destination is at very near distance and current heading angle is on the smaller left side of the destination then turn at slow rate towards right to align with the goal	1	-1	1	0
If destination is at very near distance and current heading angle is on the extreme left side of the destination then turn at very extreme rate towards right to align with the goal	1	-5	4	0
If destination is reached with current heading angle being on the smaller left side then car will stop and turn at slow rate towards right to align with the goal	0	-1	1	1

The outputs from the two neural controllers are used to make the final decision by setting their priority level in the main controller. Hurdle avoidance behavior has a higher priority in order to avoid collision with nearby obstacles around the robot. If hurdles are present in the very far region of sensors, then goal reaching behavior is activated which drives the robot towards goal by adjusting the steering angle of the robot in a smooth fashion.

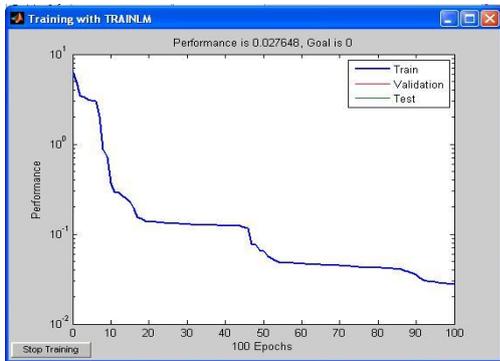


Fig. 9. Training error graph for hurdle avoidance controller

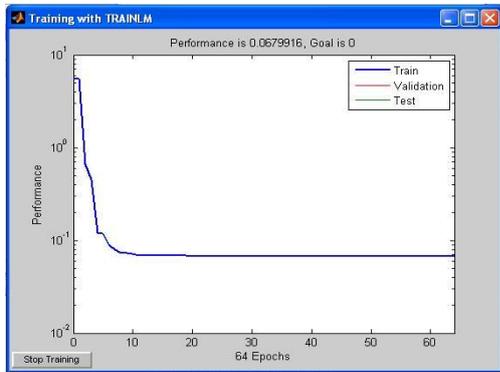


Fig. 10. Training error graph for goal reaching controller

#### IV. CONTROLLER IMPLEMENTATION AND RESULTS

After offline training in MATLAB®, the neural networks are implemented using two 89C52 microcontrollers. Keeping in view the low memory and processing power of the microcontroller, tangent-sigmoid function is converted into piecewise linear function for implementation using microcontroller and the converged weights are converted into integer form. The approximated function is described in (4) [1], [2]:

$$f(x) = \begin{cases} 0.8x & 0 \leq x < 1 \\ 0.2x + 0.6 & 1 \leq x < 1.8 \\ 0.05x + 0.87 & 1.8 \leq x < 2.5 \\ +1 & x \geq 2.5 \end{cases} \quad (4)$$

A comparison of actual tangent-sigmoid function and its approximation is shown in Fig. 11. The car with the proposed neural controller is tested in variety of environments containing obstacles and is found to reach the targets by avoiding collisions with obstacles in its way. During experimentation, the performance of the obstacle avoidance controller is found satisfactory. However, the performance of

goal reaching controller is found limited by the resolution of GPS receiver. The wheel encoder is therefore employed to estimate the car position in conjunction with data provided by GPS receiver. A test run of the car in corridor environment with obstacles is shown in Fig. 12 where it is set to reach the other end of the corridor near the standing person.

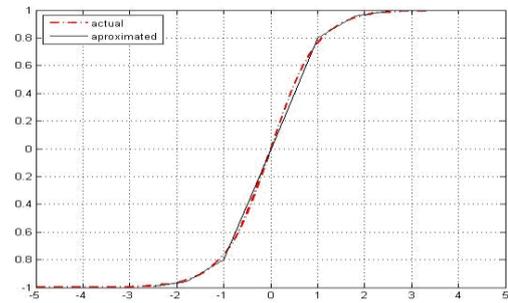


Fig. 11. Comparison of Tangent-Sigmoid Function and Approximated Function



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)

Fig. 12. (a-i) Test run of car in corridor environment where destination is in line of sight with initial position of car

## V. CONCLUSIONS

This paper describes the design of neural network based intelligent autonomous vehicle. Two neural network controllers namely hurdle avoidance and goal reaching are constructed to accomplish the navigation task. Both these controllers are feed forward neural networks trained off line with back propagation learning algorithm and implemented in real time with AT89C52 microcontrollers by using the linearized version of tangent sigmoid activation function. The testing of the controller is carried out in unknown environments and satisfactory performance is achieved. However, the use of approximated function will produce an error term which will accumulate as the number of layers will increase and efficiency of the neural controller will deteriorate further. To overcome the problem, more AT89C52 microcontrollers will be needed to run the neural controllers with actual tangent sigmoid function or DSP processor can be deployed to perform the task. The other solution is to use RAM based neural networks that do not require any activation function.

## REFERENCES

- [1] U. Farooq, M. Amar, E. ul Haq, M. U. Asad, and H. M. Atiq, "Microcontroller based neural network controlled low cost autonomous vehicle," in *Proc. International Conference on Machine Learning and Computing*, 2010, pp. 96-100.
- [2] U. Farooq, M. Amar, K. M. Hasan, K. Akhtar, M. U. Asad, and A. Iqbal, "A low cost microcontroller implementation of neural network based hurdle avoidance controller for a car-like robot," in *Proc. ICCAE*, 2010, pp. 592-597.
- [3] D. Janglova, "Neural networks in mobile robot motion," *International Journal of Advanced Robotic System*, vol. 1, no. 1, 2004, pp. 15-22.
- [4] J. A. Janet, R. Gutierrez, T. A. Chase, M. W. White, and J. C. Sutton, "Autonomous mobile robot global self localization using kohonen and region-feature neural networks," *Journal of Robotic Systems*, vol. 14, no. 4, 1997, pp. 263-282.
- [5] H. A. Awad and M. A. Al-Zorkany, "Mobile robot navigation using local model networks," *International Journal of Information Technology*, vol. 1, no. 2, pp. 58-63.
- [6] A. Chohra, A. Farah, and C. Benmehrez, "Neural navigation approach for intelligent autonomous vehicles in partially structured environments," *Applied Intelligence*, vol. 8, no. 3, May-June 1998.
- [7] R. W. Sinnott, "Virtues of the Haversine," *Sky and Telescope*, vol. 68, no. 2, 1984, p. 159.
- [8] M. H. Beale, M. T. Hagan, and H. B. Demuth, *MATLAB Neural Networks Toolbox: A User's Guide*, Mathworks Inc., 2010.



**Umar Farooq** did his B.Sc. and M.Sc. both in Electrical Engineering from University of Engineering & Technology Lahore in 2004 and 2010 respectively. He is currently with the Department of Electrical Engineering, University of The Punjab Lahore. His research interests include the application of intelligent techniques to problems in control engineering, robotics and power electronics.



**Muhammad Amar** did his B.Sc. in Electrical Engineering from University of The Punjab Lahore in 2010 and M.Sc. in Electrical Engineering from University of Engineering & Technology Lahore in 2012. He is currently working towards Ph.D. degree in Electrical Engineering from Monash University, Australia. His research interests include the application of intelligent techniques to problems in control engineering, robotics and machine vision.



**Muhammad Usman Asad** did his B.Sc. in Electrical Engineering from University of The Punjab Lahore in 2010. During his stay at Electrical Engineering Department University of The Punjab Lahore, he served as President of Society of Engineering Excellence (2009) and contributed in the research activities of the society. He is the recipient of Gold Medal award for his paper on Ball Scoring Robot in 24<sup>th</sup> IEEE International Multi-topic Symposium, 2009 and Silver Medal award for his paper on Neural Controller for Robot Navigation in 26<sup>th</sup> IEEE International Multi-topic Symposium, 2011. He is currently working towards M.Sc. degree in Electrical Engineering from G.C. University Lahore. He is with Department of Electrical Engineering, The University of Lahore where he is a Lecturer. His research interests include intelligent control of Robotics and Power systems.



**Syed Omar Saleh** holds B.Sc. degree in Electrical Engineering from University of The Punjab Lahore. During his stay at Electrical Engineering Department University of The Punjab Lahore, he served as President of Society of Engineering Excellence (2011) and contributed in research activities of the society. He won the best research paper award twice in IET All Pakistan Electrical Engineering Conferences in 2010 and 2011 held at Ghulam Ishaq Khan Institute of Engineering Sciences for his papers on Fuzzy Logic and Neural Control of Robots and silver medal in 26<sup>th</sup> IEEE International Multi-topic Symposium, 2011. His research interests include the intelligent control of Mechatronic and Power systems.



**Athar Hanif** holds B.Sc. and M.Sc. degrees in Electrical Engineering from University of Engineering & Technology Taxila and University of Engineering & Technology Lahore respectively. He is currently working towards the Ph.D. degree in Control Engineering from Muhammad Ali Jinnah University Islamabad. He is with Department of Electrical Engineering, The University of Lahore where he is working as Assistant Professor. His research interests include the robust nonlinear control of hybrid vehicles and power converters.