Controlling A Robot in Unreachable Places with GPS and Ultrasonic Sensors

Dr. Sadeq Othman AlHamouz
Faculty of Information Technology
The World Islamic Sciences & Education University W.I.S.E
P.O. Box 1101 – Amman, 11947 JORDAN.
Sadeq.Alhamouz@wise.edu.jo

Abstract -**Guiding and navigating a robot through obstacles can be integrated into many applications; such as guiding visually impaired people, or driving the robot through an unknown terrain. As such developing a reliable robot which, in addition to being remotely navigated, can analyze its' path and concludes the best-unobstructed route is required. The use of Neural Networks to analyze received data from sensors to decide on the best path provides high reliability as well as fast routing. This will also help in preserving robot's motors' power. This research implements a Festo-Robotino robot that has six ultrasonic sensors to navigate through a multi-obstacle path to decide on the best route to take. This system is also equipped with a GPS to locate it and remotely control it if necessary.**

Keywords: robot, Neural Network, Global Positioning System, ultrasonic sensor, controller.

I. INTRODUCTION

Humans are becoming more and more dependent on utilizing robots, since they have the ability to be located in areas and situations where a human being is not able to; this includes places that are impossible to reach (like the inside of a volcano), or a place that can affect man's health (like disease infected areas), or places that cost a lot to send a human being to (like outer space).

These robots need to be controlled remotely in the case the intelligent system that it is equipped with, couldn't handle a situation, or real-time actions need to be taken by a human operator. Global Positioning System (GPS) was used as a means of locating and navigating of a robot since this technology works even with the presence of having obstacles like buildings and people[1].

Robots are mostly equipped with sensors, the type of sensors a robot is equipped with depending on the objective of utilizing the robot, and the data that is needed to be collected. These sensors vary between ultrasonic [4], laser range [13], Infra-Red (IR) [5], optical sensors [8], and many others.

Mobile robots also need to be equipped with an intelligent system to handle the inputs from those sensors. How a robotic system recognizes something as an obstacle is a key point. Utilizing a Neural Network to analyze and process sensors' input has been adopted by many researchers [12]. Neural Networks were developed upon studying how a human's brain receive and analyze data. It is considered to be a tool for modeling problems that provide a solution by depicting complex relationships between input and output of the network. A Neural Network is composed of a set of connected elements called "neurons", distributed by some graph topology, these neurons "learn" from input data either in a supervised manner, where the classification rule for output is fed to the network, or in an unsupervised manner, where the network isn't fed with pre- classified output [12].

In the developed system of this research, the neural network uses a feed-forward technique to learn and decide what motor to operate and what path to take. The data gathered from the sensors are converted from analog to digital depending on a variable threshold value. The dynamic threshold provides the neural network with more time in order to redirect the robot on higher speeds, which provides more stability.

Controlling and operating the robot remotely requires having a reliable controller, choosing the right type of controller helps in making the robot's work more efficient. The most common type is the Proportional Integral Derivative controller (AKA PID controller) which is known to be effective in addressing nonlinearity, coupling, input and state saturations, but it is considered to be challenging to use in remote control applications, so researchers used "smarter" version of them, like fuzzy PID [7] where the controllers use fuzzy logic methods to set the weights for self-tuning of a basic PID controller. Another "smart" PID is the nonlinear PID controllers [10], which combines the controller with a Neural Network to overcome the linearity problem of basic PIDs.

The system developed in this research employs the Festo-Robotino (Omni-Wheel drive) robot equipped with six ultrasonic sensors distributed on the front side of the robot; upper right, upper left, upper middle, and lower right, lower left, lower middle. The feed from these sensors is analyzed by a Neural Network to recognize obstacles in its path to avoid them, in addition to the ability to adjust and navigate to GPS locations which can be directed and controlled remotely.

This research paper is divided into the following sections: the next section provides a literature review of the topics covered by this research, then a section that illustrates the basics of operating the robot initially, followed by an explanation of the neural network part of this research. Test results and experiments on this system are written in the section the following section, and finally, conclusions and suggestions to future work are stated.

II. RELATED WORK

Robotics have become a significant part of the industry these days estimated with around $ 2 billion [2]. Robots were designed and broadly used to performing repetitive and tedious tasks, such as a factory's assembly line. Then they were equipped with mobility gadgets to make more use of them.

Utilizing the Global Positioning System (GPS) as a means of locating and navigating of a robot has been adopted by many researchers due to its ability to locate and navigate robots with the presence of small and large obstacles [3]. Other means of locating and navigating a robot, or robot mounted vehicles, were used similarly to the work of [8] as Inertial Navigation System (INS), in addition to Global Positioning System (GPS).

Sensors that robots are equipped with vary depending on the purpose of utilizing that robot. Ultrasonic sensors, which collect data by sending pulses of high- frequency sounds, and compute how long it takes to receive an echo, are used to detect obstacles in the path of a robot [4]. While an unmanned aerial vehicle [8] use many types of integrated sensors, like ultrasonic and laser rangefinders to measure the distance from the ground to calculate the aerial vehicle's altitude.

Controllers of a robot are also important to consider; the most common type of controller is known as the PID controller (Proportional Integral Derivative).

Researchers like [8] adopted a multi-loop PID controller for dynamic control and generation of trajectory in their aerial robot. Fig. 1 shows the architecture of a multi-loop PIS controller.

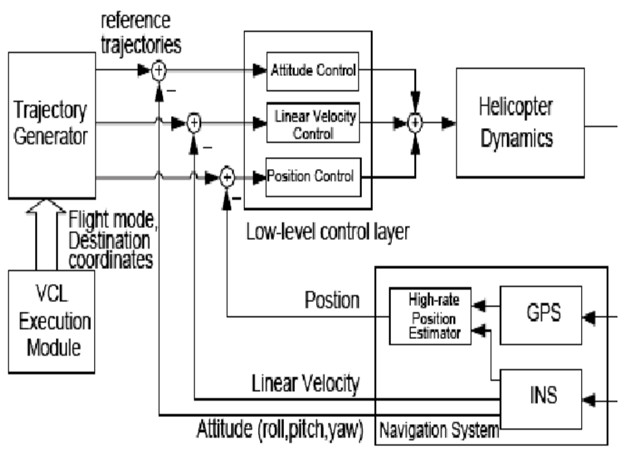

Figure 1: a multi-loop PID controller architecture (Jin Kim, et al, 2003)

Another variation of PID controllers was used in a differential drive robot in the work of [9] where they implemented a proportional integrated derivative PID controller. The robot that is used is equipped with optical encoders to help control the movement and speed of the robot's wheels by counting pulses received from optical sensors. Adjusting the wheels' speed and direction (rotating the wheels corresponding to obstacles) was controlled utilizing an adaptive PID control loop.

Analyzing the path the robot takes and recognize obstacles along the way needs a fast and reliable analysis tool. Neural Networks are considered by many researchers, to be one of the best tools for that [12].

The work of [4] presented a robot that has communication capability via GPRS. The robot uses eight ultrasonic sensors, to enable it to navigate through obstacles.
The Driving and Steering (DC) technique is governed by the microcontroller's static, pre-programmed algorithm, and the operator of the robots guides the robot remotely by utilizing a mobile phone.

Controlling a robot in off-shore areas was adopted by [6]. They used MOBOT that was controlled with remote sensing operation through DTMF signal to make a mobile robot navigate through obstacles and send images of the surrounding environment. A set of infra-red (IR) sensors were mounted on the robot to facilitate detecting obstacles, and help the operator to avoid them by sending movement commands.

[11] presented a model with predictive neural control that has a combination of a neural network feedback controller and a state-dependent Riccati equation (SDRE) controller. Yet the heavy computations needed for this analysis system makes it unfeasible in some applications like guiding a robot in an area with many obstacles.

In a motion robot, two neural networks were integrated, as in the work of [12]; one was assigned to conclude which part of the path is clear according to the feed from ultrasonic sensors, while the other Neural Network determines the safe direction of movement for the path ahead, while avoiding any "near" obstacle if existed.

III. THE ALGORITHM

The system is initiated when the power is connected to the robot and its hardware is initialized by setting and calibrating input and output ports. The internet connection is set and the robot's current position is registered with the GPS device. Once all initializations are done, the live streaming from the mounted camera begins and visuals are transmitted to the operators' device.

Now the sensors start reading the surroundings of the robots and the robot's motors begin moving. If any of the six ultrasound sensors picks a "possible" obstacle along the way, the neural network of the robot starts analyzing this reading and changes the robot's direction and the timer's Interrupt Service Routine (ISR) accordingly. Fig. 2 show the sequence of actions in the timer's ISR.

The timer interrupt keeps reading the robot's current location and calculates the remaining distance to the destination. The angle of movement is calculated and the shaft encoders change the needed motor's speed to face the angle of direction.

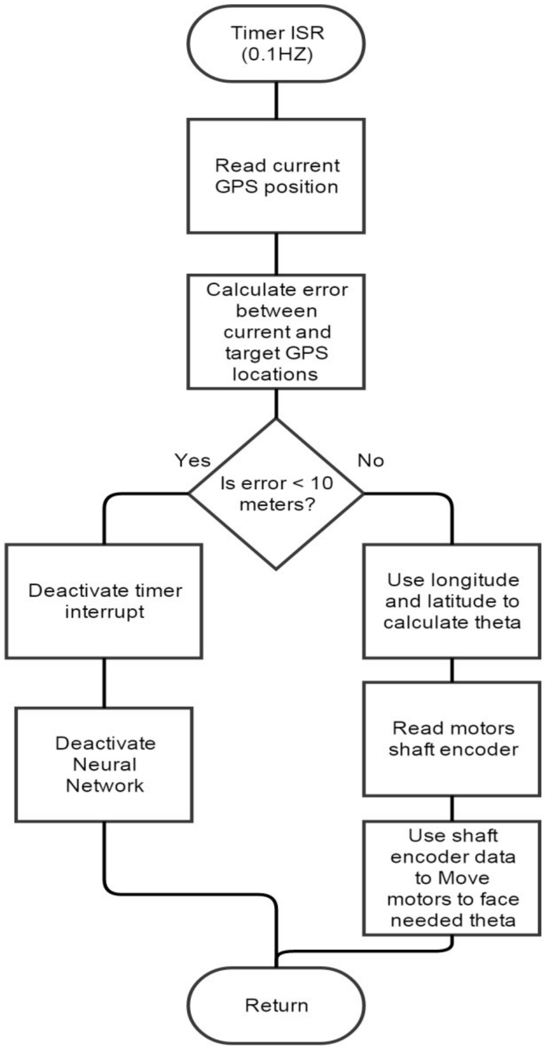

Figure 2: Flowchart of the timer's ISR

If the operator wishes to change the robot’s destination, a new GPS location is sent to the robot in an encoded form.
Once decoded by the robot, the robot sets a new path, deactivates the timer’s ISR, and moves to the new location; also stay aware of any obstacles that may be along the way in the manner described above. At any time the operator may also manually navigate.

This is done by sending navigation commands through the internet.

When the robot receives a manual navigation command, the neural network, and the ISR are deactivated, and the received command is applied instead. Fig. 3 shows a flow chart of the entire system.

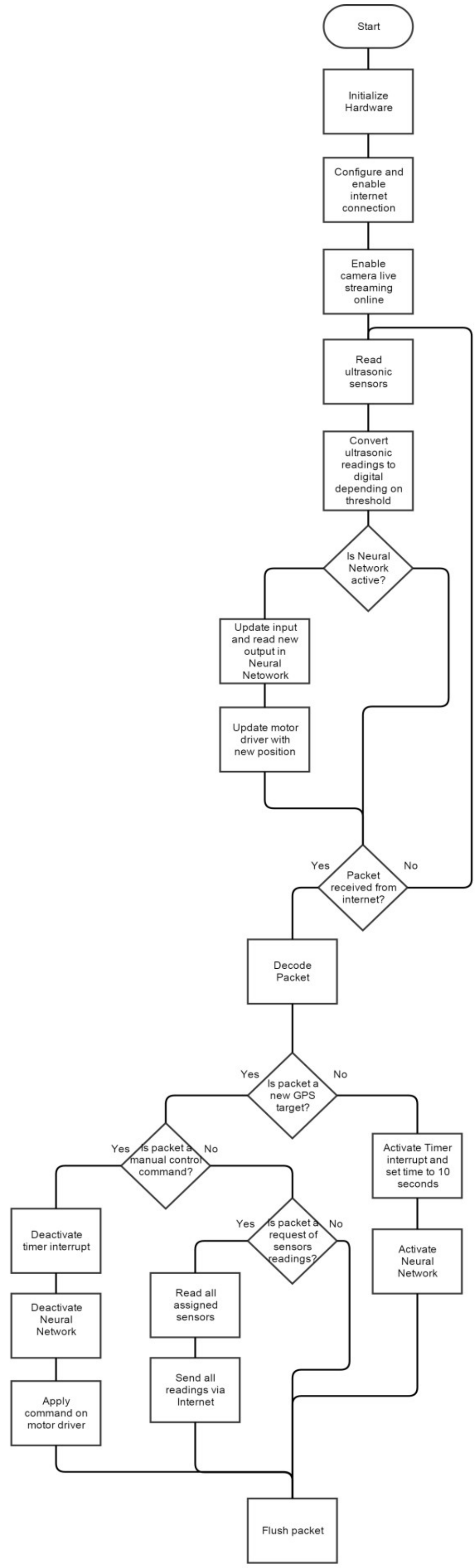

Figure 3: algorithm flowchart

IV. SETTING THE ROBOT TO WORK

We equipped a ready-made robot called Festo-Robotino (shown in Fig. 4) with six ultrasonic sensors; three in the front and three on the back. The feed from these sensors is fed into a Neural Network to be used in analyzing objects in the robot's path as obstacles and run an algorithm to avoid them; in order to reach the destination.

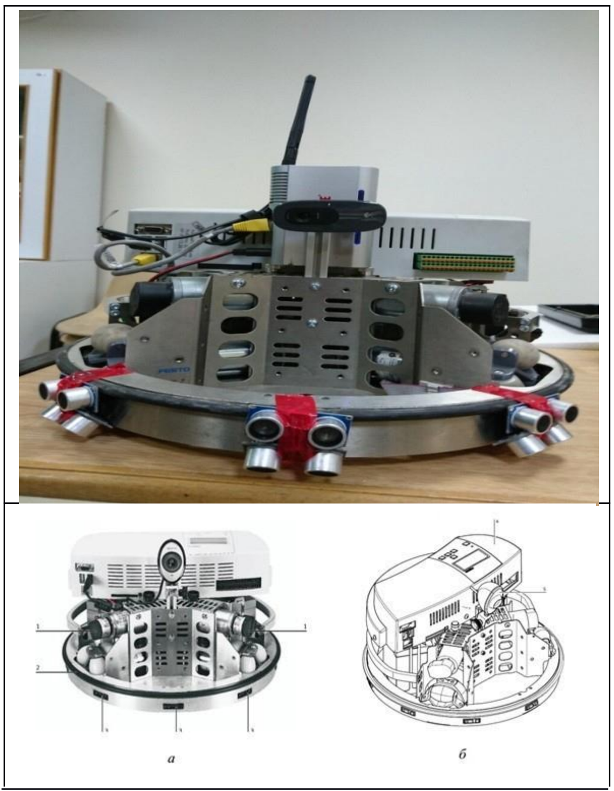

Figure 4: Festo-Robotino Robot with six ultrasonic sensors

The robot is also equipped with a camera at the front to broadcast live video to the base station (the controller's site) at a predefined rate of 25 Frames per Second (FPS). This only helps the operator "see" what is in the robot's path.

Setting the robot to work starts by initializing its hardware by setting and calibrating input and output ports, and the serial baud rate. Then the robot's embedded processor tries to configure the network by connecting to a wireless 4G router and configures a static IP address, the router has port forwarding options set to give the robot online accessibility. Fig. 5 shows the entire system as a black box, with input and output.

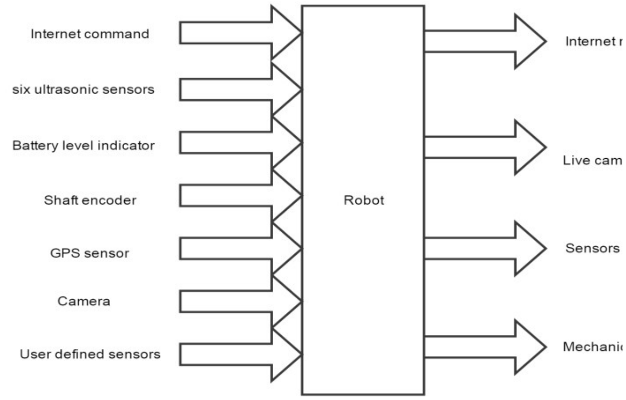

Figure 5: the system's black box

Once the robot is initialized, and all hardware is ready to operate, the ultrasonic sensors start reading data from the surrounding environment. These analog readings are converted to digital according to a variable threshold value.

The threshold (T) is obtained from a mathematical equation which relies on robot's speed. The threshold is determined by the equation:

$$T = Speed \times K$$

Where *Speed* is the current robot's speed calculated from shaft encoders feedback and has a maximum of 10 Km/h, and K is constant that is set to 0.2, The resulted threshold is the distance in meter where if an ultrasonic sensor measures a distance less than the threshold, the digital reading of a sensor is assumed to be 1 or (Object detected), Else it is assumed to be 0 or (Path clear).

The dynamic threshold provides the neural network with more time to redirect the robot on higher speeds, which provides more stability to the robot's movement.

The relation between the threshold's values with the robot's speed is shown in fig. 6.

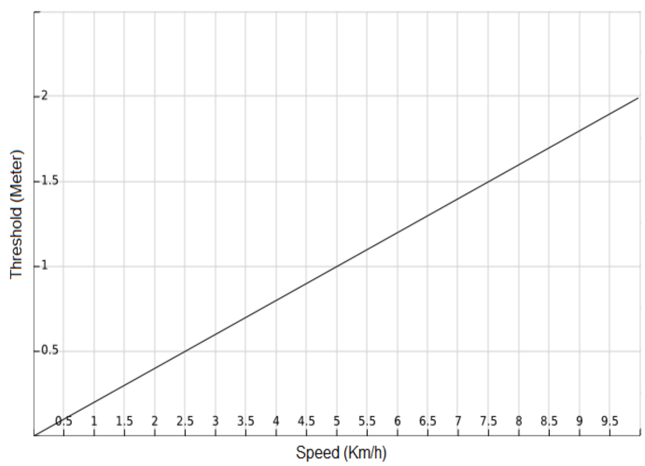

Figure 6: Threshold affected by speed change

V. THE NEURAL NETWORK

The processing of the sensor's input is performed in the Neural Network. This system employs a Back Propagation method, with four hidden layers. Fig. 7 shows the design of the Neural Network used by the robot.

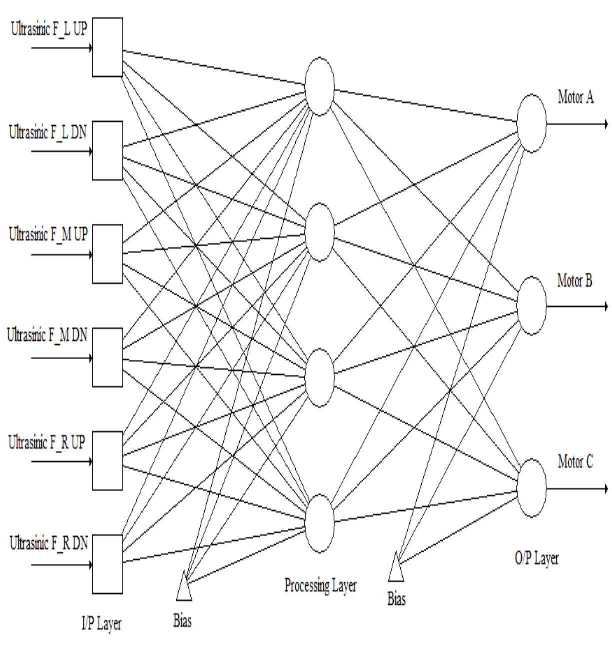

Figure 7: Neural Network design

Next, if the neural network's variable is active, the program updates the neural network's input and reads the new output, and then it uses that output as an input to the motor driver. The neural network's

output signal represents the speed and direction of robot's motors to move it and avoid obstacles.

Now the program checks if a packet was received from the internet, if there is; the packet is decoded to understand its content. If it contains a new target location, both timer interrupt and neural networks are activated. If the packet contains a manual control command such as "move forward or move left", both timer interrupt and the neural network gets deactivated in case they were active, then a command is sent to the motor's driver in order to move the robot. While, if the packet is a request to take readings from the sensors, then all basic sensors (GPS location, ultrasonic sensors readings, battery level, shaft encoders) and all user added sensors (such as temperature, humidity, LUX, magnetic field, metal detector, gas leakage, etc…), if they exist, gets readings and sends it to the user over the internet. Next, packages are flushed the program returns to reading the basic ultrasonic sensors again.

The timer interrupt occurs once every 10 seconds; it reads current longitude and latitude from GPS sensor and subtracts them from target GPS location. Assuming current robot longitude is X0 and latitude is Y0, and target longitude is X1 and latitude is Y1; the equation would be

$$X = X0 – X1$$

$$Y = Y0 – Y1;$$

Where X represents the error in longitude and Y represents the error in latitude, as shown in fig.8.

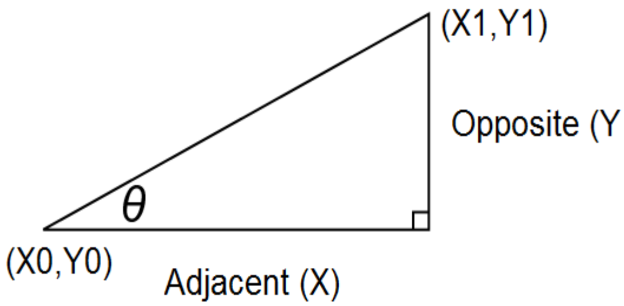

Figure 8: Calculating a new angle

If the error is less than 10 meters in longitude or latitude, it means that the robot reached its destination and thus both timer interrupt and neural network gets deactivated, otherwise, it means the robot didn't reach the destination yet, and thus θ angle can be calculated by

$$\tan^{-1}\frac{Y}{X}$$

As such the Controller uses the obtained angle to fix the orientation, then motor shaft encoders are read and the robot's position is fixed by rotating the wheels employing the motors, to the new direction.

VI. TEST RESULTS

To test the robot and put it in action, a Multi-obstacle environment was set for the robot to navigate through, as represented in Fig. 9. Sampling time = 40second / 3300sample = 12.12ms.

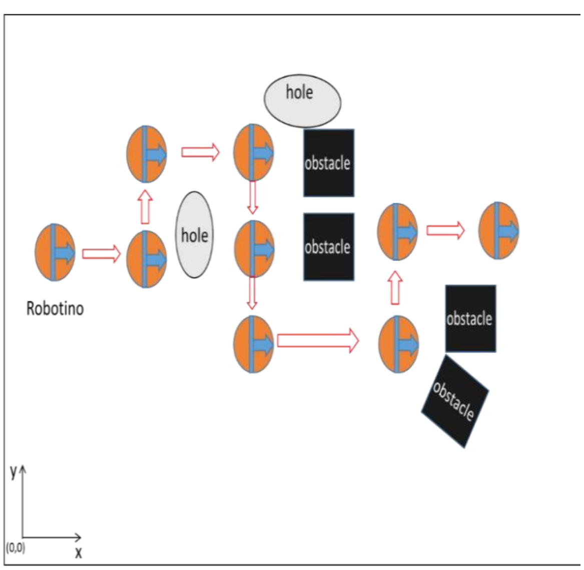

Figure 9: a multi-obstacle area to navigate the robot through

The actions that were taken by the robot to navigate were as follows:

- Start
- Go forward
- Detect a hole by the Down Front (DF) ultrasonic sensor then move Robotino left even avoid the hole, according to the NEURAL NETWORK.
- Go forward
- Detect a hole and obstacle by Down Left (DL) and Front (F) ultrasonic sensors then move Robotino right to avoid it.
- After avoiding previous obstacles Robotino detected another obstacle by Front (F) sensor as such the Robotino persists movingright.
- Go forward
- Detect two obstacles by Right (R) and Front (F) sensors so Robotino moves left.

The data that were received from the ultrasonic sensors were recorded by the neural network and were registered as shown in Fig.10:

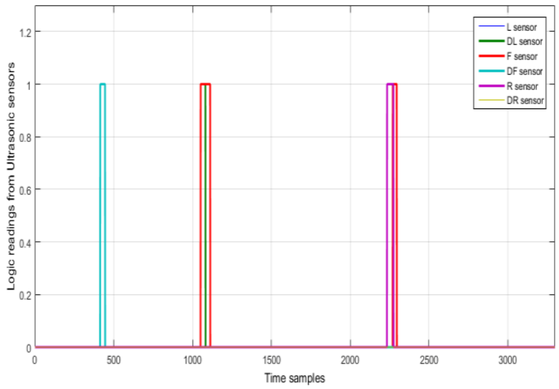

Figure 10: sensor data received by NEURAL NETWORK

We noticed that at sample 424 Down Front (DF) sensor detect a hole. And at sample 1098 Front (F) and Down left (DL) sensors detect an obstacle.

Afterward, at sample 2272 the sensor detects two obstacles by Front (F) and right (R)sensors.

The response time of the motors based on sensor readings shows that: From sample 0 to 424 Robotino was moving forward. At sample 424 the robot detects the first hole (review Fig. 9) then avoids it by moving Robotino the left. After avoiding the hole Robotino moves forward again. Then it detects the second obstacle, then the third and the fourth at sample 1098 and avoids it by moving right. At sample 2272 it detects two obstacles simultaneously then avoids them by moving to the left side. Fig. 11 below shows the response time of the motors.

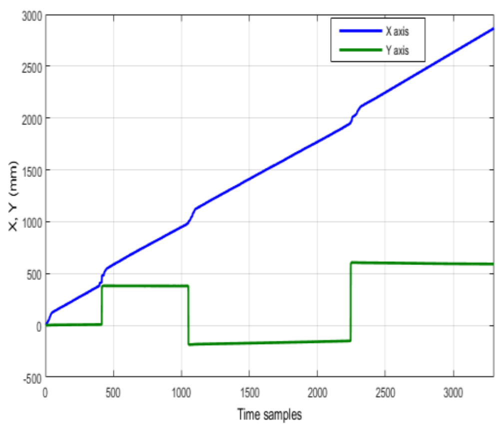

Figure 11: response time of robots' motors according to sensors readings

The actions taken by the robot in correspondence with the sensor's data are listed in detailed in table-1.

TABLE 1: ROBOT'S ACTIONS CORRESPONDING TO SENSOR'S DATA

Motor 1	Motor 2	Motor 3	Response
Forward	Forward	Forward	Rotation in clockwise direction, while stationary
Forward	Forward	Off	Rotation in clockwise direction with small radius
Forward	Forward	Backward	Rotation in clockwise direction with large radius
Forward	Off	Forward	Rotation in clockwise direction with small radius
Forward	Off	Off	Rotation in clockwise direction with mean radius
Forward	Off	Backward	Travel to 300°
Forward	Backward	Forward	Rotation in -clockwise direction with large radius
Forward	Backward	Off	Travel to 0°
Forward	Backward	Backward	Rotation in anti-clockwise direction with large radius
Off	Forward	Forward	Rotation in clockwise direction with small radius
Off	Forward	Off	Rotation in clockwise direction with mean radius
Off	Forward	Backward	Travel to 240°
Off	Off	Forward	Rotation in clockwise direction with mean radius
Off	Off	Off	Stationary
Off	Off	Backward	Rotation in anti-clockwise direction with mean radius
Off	Backward	Forward	Travel to 60°
Off	Backward	Off	Rotation in anti-clockwise direction with mean radius
Off	Backward	Backward	Rotation in anti-clockwise direction with small radius
Backward	Forward	Forward	Rotation in clockwise direction with large radius
Backward	Forward	Off	Travel to 180°
Backward	Forward	Backward	Rotation in anti clockwise direction with large radius
Backward	Off	Forward	Travel to 120°
Backward	Off	Off	Rotation in anti clockwise direction with mean radius
Backward	Off	Backward	Rotation in anti clockwise direction with small radius
Backward	Backward	Forward	Rotation in anti clockwise direction with large radius
Backward	Backward	Off	Rotation in anti clockwise direction with small radius
Backward	Backward	Backward	Rotation in anti clockwise direction, while stationary

The speed of the motors corresponds to the existence (or absence) of obstacles along the way. The speed of the back motor is almost zero because it turns ON only when Robotino is rotating, moving right or left. The right motor has a positive speed because rotate in clockwise direction and the left motor has a negative speed because rotate in an anti-clockwise direction. A graph of the changing speed of the robot’s motors is plotted in Fig. 12.

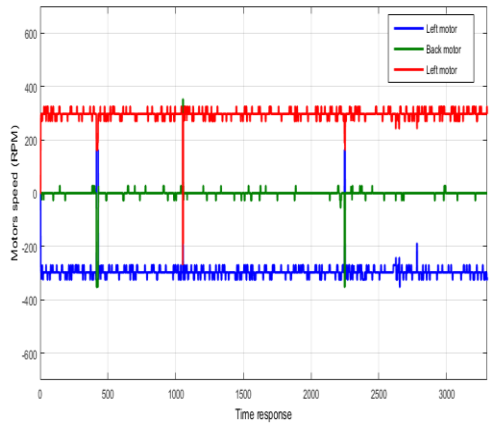

Figure 12: motors' speed when navigating through the multi- obstacle environment

Motor power consumption is considered to be very effective. Since the back motor's consumption of energy is almost zero because it only runs when the robot is moving right or left. Fig. 13 shows the power consumption of the three motors of the robot when navigating along the test path.

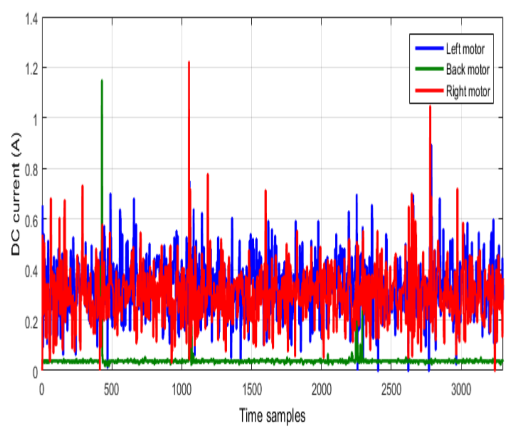

Figure 13: current signals of the three motors of the robot

The design and performance metrics of the neural network developed by this system is shown in the screen-shot of the Matlab simulator in fig. 14.

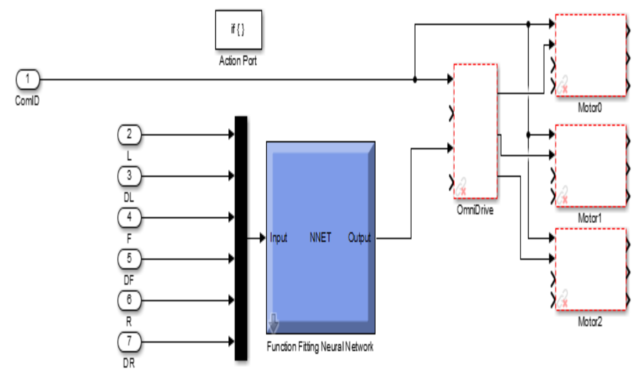

Figure 14: neural network progress data used by the Robotino system

VII. CONCLUSION AND FUTURE WORK

This research attempts to make a robotic navigation system utilizing GPS technology, with the ability to self-guide through an environment with obstacles utilizing the technology of Neural Network, which is an intelligent, yet time efficient technique that enables the robot to move safely in between obstacles. Neural Networks were proven to be an efficient technique that analyzes the sensors' readings and selects the movement for the robot's motors. Testing the robot and putting it into action in a real environment showed that this system holds promise, once put in real situations and used by ordinary people. In the future, the number of sensors could be increased to facilitate the mobility of the robot, and other types of sensors will be included and the Neural Network updated accordingly to process the diverse readings which navigate the robot through obstacles even faster and safer.

VIII. REFERENCES:

[1]. Guilherme, N. DeSouza, Avinash C. Kak, "Vision for Mobile Robot Navigation: A Survey," IEEE Transactions On Pattern Analysis And Machine Intelligence, (2002), Vol. 24, No. 2,

[2]. Siegwart, R., Nourbakhsh, I.R., and Scaramuzza, D. "Introduction to Autonomous Mobile Robots," (2011), The MIT Press, second edition.

[3]. Nakka, V. and Kabirdas, A., "Design and realization of augmented reality based navigation assistance system," Intr. J. of computer science and information technologies, Vol.2 (6), (2011), pp 2842-2846.

[4]. Al-Aubidy, K. M., Ali, M. M, Derbas, A. M. And Al-Mutairi, A. W. "GPRS-based remote sensing and teleportation of a mobile robot," 10th International Multi-Conferences on Systems Signals & Devices, (SSD13), Hammamet, . (2013), pp. 1-7.

[5]. Tas, B, Altiparmak, N, Tosun, A.,"Low-Cost Indoor Location Management for Robots Utilizing IR LEDs and an IR Camera," ACM Transaction on Sensor Network, (2014), Article 63.

[6]. Rajalakshmi, G., Celin, S., Thaj Mary Delsy, T., JamunaRani, D., "Robot for Surveillance and Implementation," International Journal of Pharmacy & Technology, (2016).

[7]. Gautam, Deepak, and Cheolkeun Ha. "Control of a quadrotor utilizing a smart self-tuning fuzzy PID controller." International Journal of Advanced Robotic Systems, (2013).

[8]. Jin Kim, H., Shim, David H., "A flight control system for aerial robots: algorithms and experiments", Control Engineering Practice 11, (2003), pp.1389–1400.

[9]. Islam, R., Shawkat, S., Bhattacharyya, S., Hossain, K., "Differential Drive Adaptive Proportional-Integral-Derivative (PID) Controlled Mobile Robot Speed Synchronization", International Journal of Computer Applications (0975 – 8887) (2014), Volume 108 – No 20.

[10]. Thanh, T.D.C., and Ahn, K.K., "Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator utilizing neural network". Mechatronics, 16(9), (2006), pp.577-587.

[11]. Wan, E. A. and Bogdanov, A. "Model predictive neural control with applications to a 6 DOF helicopter model", A IAA Guidance, Navigation, and Control Conference, Montreal, Quebec, Canada, (2001).

[12]. Janglová, D. "Neural Networks in Mobile Robot Motion", International Journal of Advanced Robotic Systems, Volume 1 Number 1, (2004), pp. 15-22.

[13]. Pomerleau, Dean A., "Neural network perception for mobile robot guidance", Springer Science and Business Media, (2012). Vol.239.