Proposed Neuro-Guided Learning for Obstacle Avoidance in AMBOA Robotic Device

Leslie R. Adrian (*Riga Technical University*)
Donato Repole (*Riga Technical University*)
L. Ribickis (*Riga Technical University*)

***Abstract* – Mobile robots, utilized increasingly in many applications, require accurate methods of obstacle avoidance. The proposed method assumes the robotic devices operational mechanism does not require data in the form of obstacle recognition, of the obstructions encountered. The exploration of relatively unknown environments including aerial, undersea, desert, icescapes or any dynamic environments in most instances require that obstacle avoidance be both automatic and autonomous and the matters of obstacle recognition (OR) may then be left to the controller, observer or to higher level systems where algorithms for visual or other recognition mechanisms may be achieved. A Guided Learning algorithm has been selected for evaluation to be incorporated within the robot system to allow high speed, memory-like reactions to its manoeuvres within chaotic, obstruction laden environments. The robotic device consists of a quad track, quad motor crawler type vehicle, purpose built to serve the function of an autonomous environmental research vehicle and as such the paper deals with the portion of the system relating to purely OA matters. The device utilizes the received sensor data from a 24 segment passive array combined with a Guided-Learning system to control the general motion of the machine through remote and unknown locations.**

I. Introduction

Obstacle Avoidance (OA) is only one element vital to the success or failure of any mobile robotic platform and from the perspective of dynamic environment navigation we must look at the main components of the overall methods adopted [3]. The following represents the overall model of the various "motion paths" consistent with most mobile robotic devices.

- Avoidance: which is absolute and refers to the need for the robot to avoid any and all obstacles excluding those obstacles where a higher system override has been invoked.
- Attraction: where the robot is programmed for a specific purpose and where pursuing a particular item or destination is the primary function. An example of this would be the photovore, (light seeking robot) designed to track and follow light or a line-tracking robot designed to follow a line or some other markers magnetic, RF etc, to a specific destination.
- Further categories of the "motion path" are visual, mapping and planned, all of which involve complicated and verbose algorithms and large memory availability and do not relate to the current paper.

The selected robot for evaluation purposes is the Ambient Obstacle Avoidance (AMBOA) robot which is equipped with a purely passive sensing system [1], in other words it relies only on received spectral wavelength and does not emit radiation such as is the case with infrared or sonic devices. The sensor array registers only wavelengths from the spectral field from the ultraviolet to the near infrared in a combination digital and analogue array. The predominant manner of programming the robot has hitherto been the use of fuzzy logic algorithms which has proven quite satisfactory however the paper takes a theoretical and logical look at another method of obstacle avoidance. The programming of the fuzzy algorithm for this task can become quite overwhelming as can be seen in the line drawing network in Fig.1. The enormity of the number of "*if-and-and-then statements*" required although quite doable is both a challenging task and would lead to many error adjustments with the number of network connections and combinations required for a 24 sensor input and 4 drive motor output. [2][3]. As such it was decided to investigate a neural learning network design as an option. After evaluation by the authors it was decided to utilize the Delta rule and an associated guided learning method as an entry level investigation as described within Section II.

II. Network Training Method

Data collection is achieved as the robot is guided via remote control (RC), through an area defined as the "selected environment", which is either the actual environment in which the robot will operate or is a near facsimile of that environment. Therefore appropriate hardware is required to provide the RC aspect of the learning process. The operator guides the robot through a series of obstacles, approaching as many as possible obstacles from all possible angles to establish a base and bias pattern for the algorithm. The onboard data collection algorithm as described in Section VII has been designed to capture both digital and analogue readings from the 24 sensor array during the allocated learning period $T_{learn,}$ during which time the received data is stored within the onboard memory chip. T_{learn} must not exceed MOB maximum as in formula (1).

$$T_{learn} < \left(\frac{(MOB\ in\ Mb)\times 2^{20}}{((Sen_d\times 1)+(Sen_a\times 2))\times S_{ps}\times S}\right) \tag{1}$$

Where:

T_{learn} = Maximum run time.
Sen_d = Digital Sensors @ 1 byte per sample.

ISBN 978-1-4673-9752-0

Sen_a = Analog Sensors @ 2 bytes per sample.
S_{ps} = Samples per second.
S = Number of seconds.
MOB = Onboard memory Mbytes.

In our configuration the robot will read the sensor array 10 times per second for 30 minutes of RC driving. Therefore, 24 sensors @ 1 byte per sample plus 24 sensors @ 2 bytes per sample multiplied by 10 samples per second multiplied by 60 seconds, represents data capture per minute approximating 43,2Kb per minute of RC navigation. With an onboard Memory of 2Mbit a maximum data collection time of around 45 minutes may be expected, though varying T_{learn} periods can be performed to determine results, the example illustrates the method by which the $T_{learn} < MEM_{max}$ may be obtained.

III. DELTA RULE AND SINGLE LAYER PROPAGATION

The data collection as outlined in Section II consists of raw sensor data from the 24 sensor array and is thereafter processed through the selected neural algorithm which in our test case is the Aforge.net C# framework which is purpose designed for developers and researchers in the fields of Artificial Intelligence, neural networks, genetic algorithms, machine learning and robotics among other things. The process is referred to as batch learning because after the data has been collected it is analysed using the Delta Rule method. As shown in Fig.1, a single layer propagation network is used as a feed forward perceptron and although many combinations exist, it still retains the classification of a basic neural network consisting of a single layer of 4 output nodes where the inputs are fed directly to the outputs via a series of 96 weights. It should be noted that there is no back propagation in the network and hidden layers if any may be generated within a given algorithm. The Delta Rule as seen in Formula 2, in its simplest form as described by [6].

$$\Delta_{w_{ji}} = -\varepsilon \frac{\delta E}{\delta w_{ji}} = \varepsilon \delta_{w_{ji}} \tag{2}$$

In its simplest form from Formula 2 it can be seen that the change in any particular weight is equal to the products of:

- the learning rate epsilon.
- the difference between the target and actual activation of the output node δ.
- the activation of the input node associated with the weight in question.

A higher value for ε will necessarily result in a greater magnitude of change. Because each weight update can reduce error only slightly many iterations are required in order to satisfactorily minimize error. In batch mode the value of

$$\frac{\delta E_p}{\delta w_{ji}} \tag{3}$$

is calculated after each sample is submitted to the network with the total derivative (4) calculated at the end of an iteration by summing the individual pattern derivatives. Only after this value is calculated are the weights updated. As long

$$\frac{\delta E}{\delta w_{ji}} \tag{4}$$

as the learning rate epsilon is small, batch mode approximates gradient descent [7].

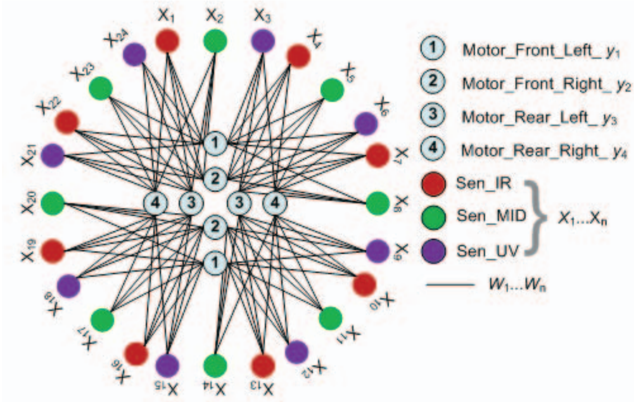


Fig.1. The network contains 96 weighted connections to 4 drive motors.

The network represented in Fig.1, is not a complete representation of the sensor input system of the AMBOA robot. While the sensor array has 24 passive sensors divided into three sensor types, Infrared (IR), Visible Spectrum (MID) and Ultraviolet (UV), the circuitry has been designed to provide both digital and analog data simultaneously for each sensor, providing a duality in the neural learning process as the Delta Rule method allows also for non-binary values. As can be seen from Formula 2 the Delta Rule is essentially a simple linear sum of products (which is represented by the symbol in the four output nodes in Fig. 2) and is used as the activation function at the output node of the network shown here.

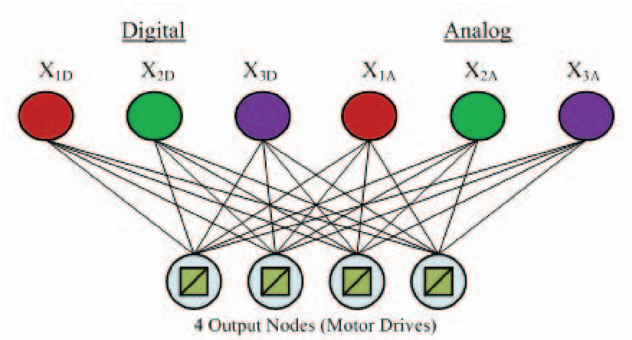


Fig.2. Three sensor input showing digital and analog inputs effectively doubling the efficiency of the network in its ability to adapt.

IV. ERROR REDUCTION

The method to be utilized as stated in Section II is referred to as guided learning utilizing batch processing, which is essentially the same as supervised learning except that in supervised learning the process results from sampling data strings one at a time and batch refers to the collection of all data and processing the samples all at once. According to examples as described by [6] which state. “With repeated presentation of the same training data to the network (with multiple iterations of training), it becomes clear that the network’s weights do indeed evolve to reduce classification error: error is eliminated altogether by the twentieth iteration. The network has learned to classify all training cases correctly, and is now ready to be used on new data whose relations between inputs and desired outputs generally match those of the training data”.

The example given within their paper revolves around a 4 node and 1 output network, with original weights set to “0” with arbitrary weight progression set to 0.25 increments. When using the Delta Rule as in formula 2 this error free result is possible with the required condition that all solutions must be a linear function of the inputs.

V. CORRELATION MATRICES

After the recorded batch data is trained within the algorithm the resultant connection strengths between the 24 input and 4 outputs are represented as a correlation matrix with associated values. The test matrix is both complex and large and as such is not able to be included herein, however the representation of the working environment of the Delta Rule frontend gives a more rounded idea of the functions able to be generated. When allowing for a 4x24/1x4 matrix from 192 connections it was found preferable to set Iterations to infinity. The results achieved from the new sensor array as defined in Section VI are expected to represent a more error resistant matrix.

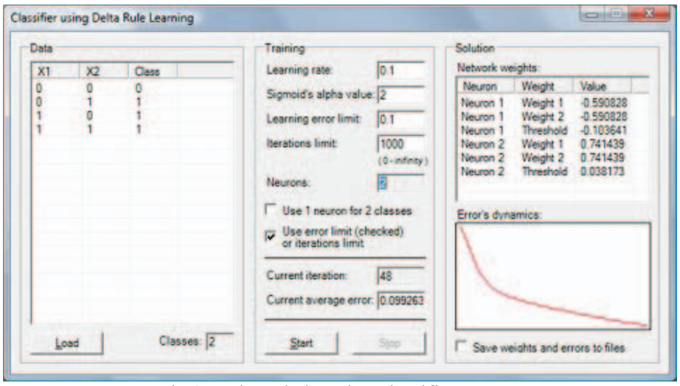
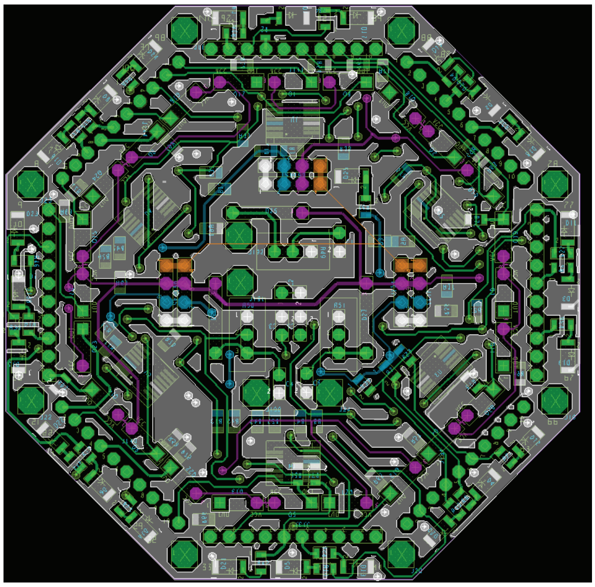


Fig.3. Delta rule learning classifier screen.

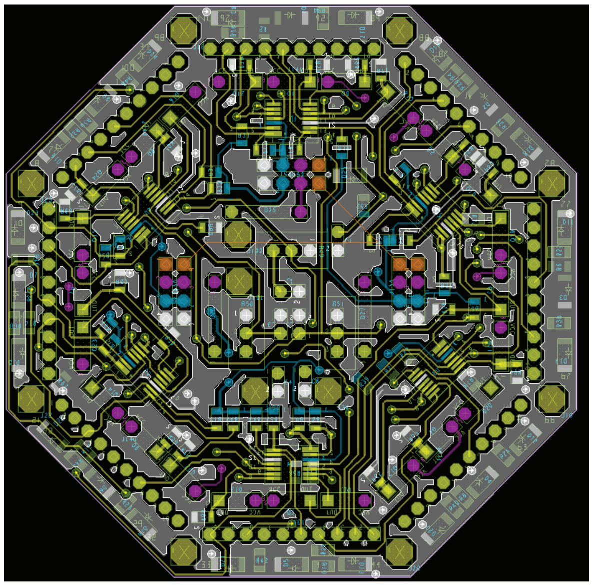
VI. AMBOA PREPARATION

Fig. 4. AMBOA frame without sensor array.

The latest AMBOA system in Fig.4, has been constructed and is awaiting finalization of the PCBs shown in Fig.5a) and Fig.5b) and have been designed with octagonal geometry so that each of the 24 sensors faces a particular quadrant. Those quadrants are specifically named for the purpose of programming references and are: Front, F, Rear, R, Left L, Right R, LF, LR, RF and RR respectively.



a)



b)

Fig. 5. Sensor wafer PCB, a) board front and b) board back.

VII. ALGORITHM FEATURES

In order to obtain the goal of a robust system at low cost, certain sacrifices have been made relating to the computational speed of the algorithm. The primary initiative is based upon shifting the load weight from the algorithm and placing reliance on the results achieved from the Delta Rule neural trainer. In other words reduce the processing cycles with respect to controllers based on conventional approaches and give to each cycle more importance upon the information processed through the use of analog based neural networks. The importance of those neural networks comes from the training on which it is developed. Training neural fuzzy algorithms means tuning the behavior function of what it is able to learn. This means it is possible to change main robot task by changing the neural fuzzy logic weights without change of algorithm. As can be seen in the flow chart of Figure 7 algorithm, except for the neural fuzzy controller block, there is no need to develop any unconventional function, it is necessary to correctly adapt for the proposed algorithm with many auxiliary functions readily available from the manufacturer such as, PWM function, SPI function, digital sensing function, analog sensing function and digital outputs.

Table 1: Data storage algorithm.

```
Main {
System Init_ ;
_delay32(16) ;
While (1)
{
Cycle = Cycle + 1;
Void Analog_Sensing_Function_(void) ;
_delay32(16);
Void Digital_Sensing_Function_ (void);
_delay32(16);
Void SPI_CH2_data_Storage_ ;
_delay_ (32000);
If (cycle > 16 000 000 ){
While(1){
};
};
};
```

Table 2: Autonomous driving algorithm.

```
Main {
System Init_;
_delay32(16);
While (1)
{
Void Analog_Sensing_Function_( void);
_delay32(16);
Void Digital_Sensing_Function_(void);
_delay32(16);
Void Neuro_Fuzzy_Control_Funct_(void);
_delay32(16);
Void Robot_Motor_Control(Void);
_delay32(16);
Void Robot_Operations(void);
_delay32(16)
Void SPI_CH1_WiFi__Communication_ (void);
_delay32(16);
Void SPI_CH2_data_Storage_(void);
_delay32(32000);
};
};
```

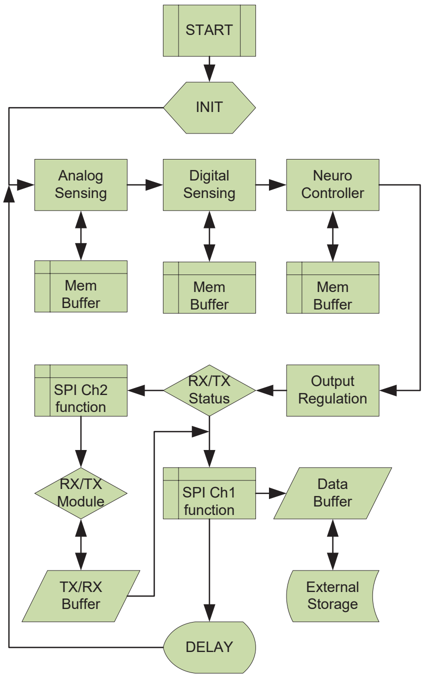


Fig.6. System flow chart.

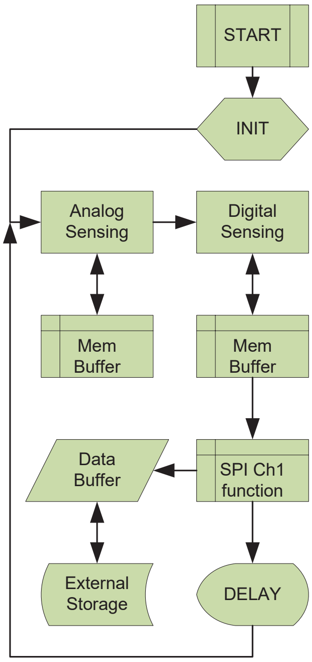


Fig. 7. Guided learning flow chart.

VIII. CONCLUSION

The results of those various methods tested, combined with research of results obtained from other references have indicated that results should be more than satisfactory and are indicative that the methods proposed are sound and indicative of more intensive research. As can be seen in Section VI the necessary PCBs have been produced and work continues to complete those parts. Many experiments have been performed using the original AMBOA sensor array however that specific array was never designed for this system. Completed construction of the prototype will allow accurate definitions of the mathematical allowances needed for the correct evaluation of the Delta Rule system in comparison to our previously performed fuzzy logic analysis [1]. Future work will continue with focus on obtaining completion of this work which adds to the efficacy of autonomous robots in open environments.

IX. Acknowledgement

"This work has been supported by Latvian national research programme LATENERGI"

References

[1] L. R. Adrian and L. Ribickis," Fuzzy Logic Analysis of Photovoltaic Data for Obstacle Avoidance or Mapping Robot", The 16th International Conference ELECTRONICS'2012, Palanga, Lithuania ,18th - 20th June 2012. No. 1(127), Jan 2013 Vol 19, No 1 (2013).

[2] N. A. V. Rao and S. S. Iyengar, "Autonomous robot navigation in unknown terrains: Incidental learning and environment exploration," IEEE Trans. Syst., Man, Cybern., vol. 20, pp. 1443–1449, Nov./Dec. 1990.Piezo Film Sensors Technical Manual - Tufts University. www.tufts.edu/programs/mma/emid/piezo.pdf

[3] C. Ye, "Behavior-based fuzzy navigator of mobile vehicle in unknown and dynamically changing environment," Ph.D. dissertation, Dept. EEE, University of Hong Kong, Hong Kong, 1999.Monika Jain, Usha Tiwari and Mohit Gupta, Mobile Charger Via Walk. IMPACT-(2011)

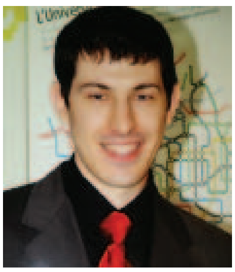
[4] N. A. V. Rao and S. S. Iyengar, "Autonomous robot navigation in unknown terrains: Incidental learning and environment exploration," IEEE Trans. Syst., Man, Cybern., vol. 20, pp. 1443–1449, Nov./Dec. 1990.

[5] L. R. Adrian and L. Ribickis, "Fuzzy Logic Control of Photo-voltaic Sensors for Obstacle Avoidance or Mapping Robot", (IJAS) International Conference for Academic Disciplines in Gottenheim, Germany. Academic Journal of Science, Vol. 1, No. 2 Dec 29, 2012.

[6] McClelland, J. L. & Rumelhart, D. E. (1988). A simulation-based tutorial system for exploring parallel distributed processing. Behavior Research Methods, Instruments & Computers, 2, 263-275.

[7] Reed, R. D., & Marks, R. J. (1999). Neural Smithing: supervised learning in feed forward artificial neural networks. Cambridge, MA: MIT Press

Leslie Adrian received his Bachelors Diploma and Master's Diploma at Riga Technical University, within the Faculties of Electronics and Telecommunications (FET) and Power and Electrical Engineering, Department of Power Electronics and Electrical Technologies (EEF).



Donato Repole BSc in University of Salerno (Italy) in Electronic Engineering and Master's Diploma at Riga Technical University, within the Faculty Power and Electrical Engineering, Department of Power Electronics Electrical Technologies (EEF).