Research Journal of Applied Sciences, Engineering and Technology 4(15): 2577-2583, 2012
ISSN: 2040-7467

Submitted: April 04, 2012 Accepted: May 22, 21012 Published: August 01, 2012

A Novel Mobile Robot Navigation System Using Neuro-Fuzzy Rule-Based Optimization Technique

Ayman AbuBaker
Department of Electrical and Computer Engineering, Faculty of Engineering, Applied Science University, Amman, 11931, Jordan

Abstract: A new novel approach to control the autonomous mobile robot that moved along a collision free trajectory until it reaches its target is proposed in this study. The approach taken here utilizes a hybrid neuro-fuzzy method where the neural network effectively chooses the optimum number of activation rules in order to reduce computational time for real-time applications. Initially, a classical fuzzy logic controller has been constructed for the path planning problem. The inference engine required 625 if-then rules for its implementation. Then the neural network is implemented to choose the optimum number of the activation rules based on the input crisp values. Simulation experiments were conducted to test the performance of the developed controller and the results proved that the approach to be practical for real time applications. The proposed neuro-fuzzy optimization controller is evaluated subjectively and objectively with other fuzzy approaches and also the processing time is taken in consideration.

Keywords: Fuzzy logic, mobile robot, neural network, rule-based optimization

INTRODUCTION

The mobile robot is constructed as manipulator for auto sensing, which is capable to navigate in an unknown topology with moving and stationary obstacles. In the design of this kind of autonomous system, two important aspects need to be addressed carefully (Aguirre and Gonzalez, 2000; Tanaka, 1997). The first of which is the design of a nonlinear and non-analytical controller, whereas the second deals with the reliability of such a system. It can be stated that human experience represented by a set of linguistic rules can be a sensible solution to this kind of control problems. Such controllers are called fuzzy logic controllers, which are designed to mimic human experience (Cao *et al*., 1999; Yung and Ye, 1999). Although, fuzzy logic controllers are robust, but deriving and fine-tuning the entire rule set of a fuzzy logic controller is a tedious and difficult problem. Furthermore, fuzzy controllers can be very slow in response when used for large-scale problems (Ross, 1998; Zadeh, 1997).

Among all the soft-computing methods suggested for mobile robot reactive navigation, fuzzy logic systems have been found to be the most attractive. They are tolerant to noise and error in the sense of information coming from the sensory system and most importantly they are factual reflection of the behaviour of human expertise. In general, there are two approaches to the application of fuzzy logic in mobile robot navigation, namely, behaviour-based approach (Aguirre and Gonzalez, 2000; Harisha *et al*., 2008; Pradhana *et al*., 2009) and classical fuzzy rule-based approach (Parasuraman *et al*., 2008; Ross, 1998; Zadeh, 1997). The design of fuzzy logic rules is often reliant on heuristic experience and it lacks systematic methodology. Therefore this rules might not be correct and consistent, do not possess complete domain knowledge and/or could have a proportion of redundant rules. Furthermore, when a better precision is needed the number of input variables and their fuzzy values need to be increased, for example, when using four input variables each mapped by seven fuzzy values besides 2401 if-then rules may be required to define the rule-base of the inference system. Such huge expansion in a multi-dimensional fuzzy rule-based system adds further ad hoc to the design of the system (Tanaka, 1997).

Several successful reactive navigation approaches based on neural networks have been suggested in the literature (Callan, 1999; Carpenter *et al*., 1992; Fausett, 1993). In spite of various suggested network topologies and learning methods, neural reactive navigators still perceive their knowledge and skills from demonstrating actions. Therefore, they suffer from a very slow convergence, lack of generalization due to limited patterns to represent complicated environments and finally information encapsulated within the network cannot be interpreted into physical knowledge (Singh *et al*., 2008). Consequently, the utilization of neural networks in reactive mobile robot navigation is limited when compared to fuzzy logic. However, the role of neural networks has been found to be very useful and effective when integrated with fuzzy systems (Cao *et al*., 1999; Yung and Ye, 1999). The birth of this integration between these two soft-computing paradigms is the neuro-fuzzy systems. Neuro-fuzzy systems provide an urgent synergy that can be found between the two paradigms, especially the capability to mimic human experts as in fuzzy logic and learning from previous experience capability as in neural networks.

In general, neuro-fuzzy systems can be classified into three categories, neutrally adaptive fuzzy inference system, neutrally performed FIS and combinatorial, or hybrid, neuro-fuzzy systems. The neutrally adaptive fuzzy inference system is the most widely used neuro-fuzzy systems and they are designed to combine the learning capabilities of neural networks and reasoning properties of fuzzy logic (Medsker, 1995).

In this study, a new approach is proposed to design a neuro-fuzzy rule base optimization technique that can control the mobile robot in unstructured environment. The proposed system has apparent advantage in structures that simplify and reduce the processing time and improve the performance. This archived by applying the neural network to activate the optimum number of activation rule base in the fuzzy inference engine. With such a technique, the required time needed to infer the decision for the robot movement is extremely reduced.

Examples of related published material: Previous authors have applied various types of techniques to control the mobile robot using different intelligent techniques. Aguirre and Gonzalez (2000) developed and successfully implemented a pattern recognition approach to reactive navigation based on real time sensory information. A heuristic fuzzy neuro network is developed for pattern-mapping between quantized ultrasonic sensory data and the velocity commands to the robot. The design goal was to enable an autonomous mobile robot to navigate safely and efficiently to target position in a previously unknown environment. To build the desired mapping between perception and motion, usefully heuristic rules were combined with the Fuzzy Kohonen Clustering Network (FKCN). In Yung and Ye (1999) study, an alternative training approach to the EEM based training method is presented and fuzzy reactive navigation architecture is described. Using the rule base learned from the new method, the proposed fuzzy reactive navigator fuses the obstacle avoidance behaviour and goal seeking behaviour to determine its control actions The new training method is 270 times faster in learning speed; and is only 4% of the learning cost of the EEM method, where adaptability is achieved with the aid of an environment evaluator. Hagras and Sobh (2002) discussed the control of autonomous intelligent robotic agent operating in unstructured changing environments. Online learning as a useful method producing intelligent machines for inaccessible environments was introduced. In these environments it is required to perform online learning through interaction with the real environment and performing any adaptation within short time intervals. Under such conditions, robotic agents have to be adaptive. Cao *et al*. (1999) describe a Neuro-fuzzy control method for the navigation of an AGV robot. An overall system design and development was presented. The Neuro-Fuzzy computation and its application for mobile robot navigation were discussed. The system that was to be controlled is an electrically propelled mobile vehicle named Bearcat II, which is a computer controlled intelligent system. As autonomous navigation requires a number of heterogeneous capabilities, including the ability to execute elementary goal-achieving actions, like reaching a given location; to reach in real time to unexpected events, like the sudden appearance of an obstacle; to determine the robot's position; and to adapt to changes in the environment, the study introduces a Neuro-fuzzy control method for navigation of an Autonomous Guided Vehicle (AGV) robot. Ng and Trivedi (1998) introduce a Neural integrated Fuzzy controller (NiF-T), which integrates the fuzzy logic representation of human knowledge with the learning capability of neural networks, is developed for nonlinear dynamic control problems. It covers integrated sensing, control, actuator modules, real-time performance, ability to successfully handle noisy sensor signals, reactive controller design which captures high-level, linguistically based human expertise in a set of fuzzy rules, training of neural networks directly with fuzzy rules instead of numerical sample data, as well as learning capabilities and general applicability. NiF-T architecture comprises "of three distinct parts:

- Fuzzy logic Membership Functions (FMF)
- A Rule Neural Network (RNN)
- And Output-Refinement Neural Network (ORNN)

FMF are utilized to fuzzy sensory inputs. RNN interpolates the fuzzy rule set; after defuzzification, the output is used to train ORNN. The weights of the ORNN can be adjusted on-line to fine-tune the controller". Only five rules were used to train the wall following behaviour, while nine were used for the hall cantering. Also, a robot convoying behaviour was realized with only nine rules. For all of the described behaviors-wall following, hall centering and convoying, their RNN's were trained only for a few hundred iterations and so are their ORNN's trained for only less than one hundred iterations to learn their parent rule sets.

The proposed fuzzy navigation system: The mobile robot is required to explore several paths in a maze, of a pattern of successive combination

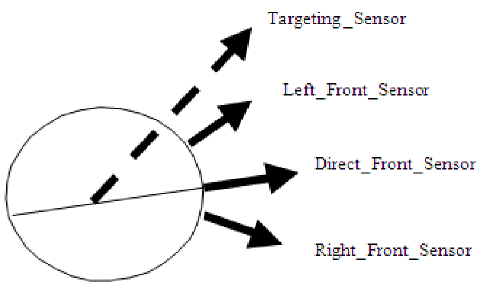


Fig. 1: Mobile robot with frontal sensors

of left and right turns. Its task is to reach a desired position at the end of one channel. The mobile robot uses a kind process, sequentially adopting cyclic pattern of the left and right turns. Eventually, it ends up with the desired position, at which time a signal is injected, causing the robot to record the correct pattern. The mobile robot is assumed to be equipped with three physical ultrasonic sensors and one virtual sensor as shown in Fig. 1.

The physical sensors are used to detect obstacles in front of the robot, the right side and the left side, respectively. The maximum distance that can be sensed by these sensors is assumed to be 6 m. The virtual sensor is used to guide the robot towards the target. This sensor is especially needed when the target direction of movement is totally blocked by an obstacle. The virtual sensor will guide the robot back towards the target once the obstacle is avoided.

Henceforth, the robot travels quickly and accurately along the track to accomplish any job that has been assigned. It is assumed that the robot will not face any traps (or get into a situation where it is required to backtrack or turn around). Such a problem is out of this study scope.

The four sensors provide the path planning system of the robot with three distances front (dc), right (dr), left (dl) and target orientation (theta), respectively. From these inputs, the fuzzy logic controller will make up a decision in which direction should the robot move in order to reach the target. The Neuro-Fuzzy Optimizing Controller (NFOC) is divided to four stages as shown in Fig. 2.

Forty rules fuzzy navigator system: The Fuzzy Logic Controller (FLC40) was analyzed and tested for different cases based on the same parameters and rules used by Xu and Tso (1996). The robot motion results have been considered with relation to different cases.

Problems were recorded and investigated and the reasons behind the failure of this robot, in these cases, were related to the limited number of the sets used (FAR, NEAR) and the limited angle of orientation (turning angle), which are five sets. Due to this limitation, the robot touches the obstacles slightly in all cases considered as shown in Fig. 3. To avoid these problems, a relaxation of the rules was done by increasing the number of sets for the input distances from 2 to 5 sets; accordingly, the number of rules was increased to 625 activation rules which will be discussed in the next section.

Development of the improved fuzzy navigator system: As it has been already noted, the FLC 40 is not capable to avoid collision with the edges of the obstacles in all cases. The main reason behind that failure is the low resolution due to two fuzzy sets, i.e., FAR and NEAR. An improvement to the system can be easily made by increasing the number of fuzzy sets in order to achieve better resolution. In this study, it is proposed to increase the fuzzy sets to five linguistic labels (VL, L, M, S, VS) as shown in Fig. 4a to c. The fuzzy sets in this case become shorter than before, so the accuracy and the performance of the controller are improved. As the number of sets is increased the fuzzy rules are increased as well up to 625 activation rules ($5\times5\times5\times5 = 625$ activation rules).

The results obtained from this improved fuzzy logic controller have been improved. The robot avoids collision with the obstacles as shown in Fig. 5, but the main problem in using that improved controller is the processing time. It is very long, since the number of rules is high and requires more time to create a decision and this will affect the response time of the robot.

Neuro-fuzzy optimizing controller: The main problem in the fuzzy logic controller is the inference block,

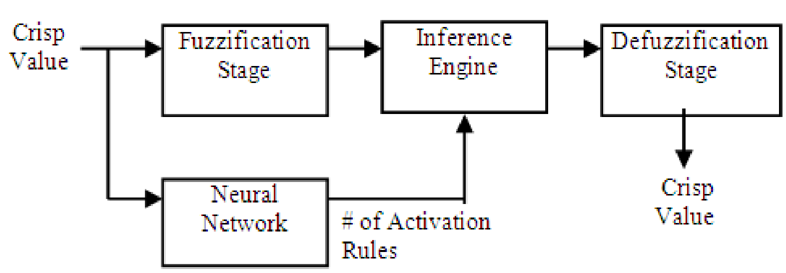


Fig. 2: NFOC flowchart

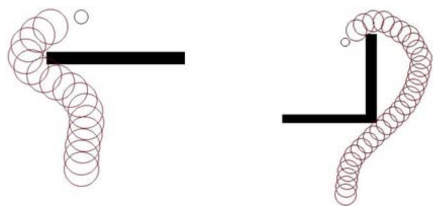


Fig. 3: FLC40, simulated motion

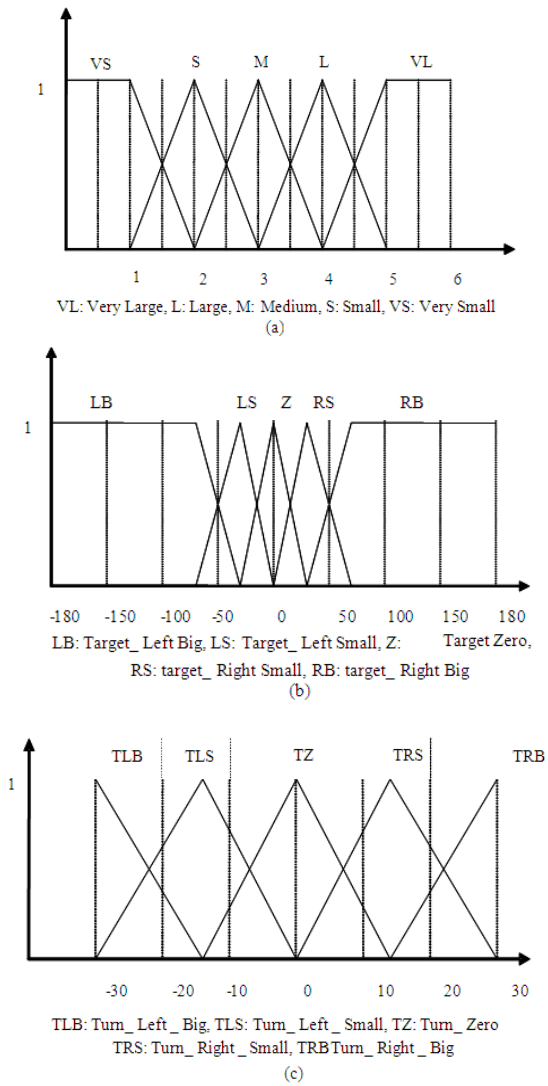


Fig. 4: (a) Distance, (b) target orientation, (c) turning angle membership function

which consists of a large number of rules that need a long processing time. To solve this problem of processing time, the Neuro-Fuzzy Optimizing Controller (NFOC) is proposed, where neural network will be utilized to choose the optimum number of the activation

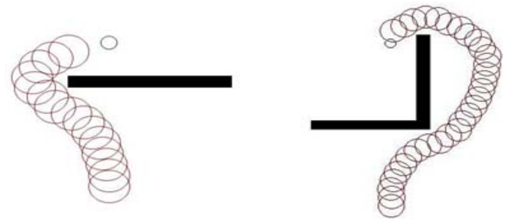


Fig. 5: Improvement FLC625 success cases

rules in the inference engine. The system is investigated by considering the results of the integration between both systems (Fuzzy logic and neural networks) as shown in Fig. 2.

Creating the input and the output data: The neural network training in the feed forward method, the inputs and the outputs data must be known. The inputs data are crisp value for the three distance sensor and the physical orientation sensor. As shown in Fig. 4, the distance range is [0-6] meters, but the interval from 1-5 is considered since the distance interval [0, 1] has the linguistic label as 1 m (VS) and the distance interval [5, 6] has also the linguistic label 5 m (VL). The same thing is for the angle orientation in which the interval that considered is [-65, 65] degrees since the sided sets have the linguistic labels as -65 and 65, respectively. The distance is divided into seventeen points with step of 0.25 m between each point and step of 8.125 degree for the orientation angle input. So the resulted matrix for each input will be 17 rows with 4 columns corresponding to the number of input sensors. As a result of the combinations of all these four matrixes, the input matrix of the neural network will be 83521 rows with 4 columns where the number 83521 comes from the combination of the all matrix rows $17\times17\times17\times17 = 83521$ and the number 4 comes from the 3 ultrasound distance sensors and 1 physical orientation sensor. To find the output, the fuzzy rules in the inference engine are divided to 16 sections. Each section includes 40 rules.

As an example, a sample is presented where the activation rules are:

IF dr is L and dc is L and dl is L and tr is LB THEN Sa is TLB
IF dr is S and dc is M and dl is L and tr is RB THEN Sa is TZ
IF dr is L and dc is M and dl is S and tr is RS THEN Sa is TRS

As a result, the total number of rules will be 640 activation rules. Where the 625 will actual activation rules and 15 do nothing rules that will be used null action cases. Therefore, the output nodes for the neural network are set to sixteen nodes which are the number of

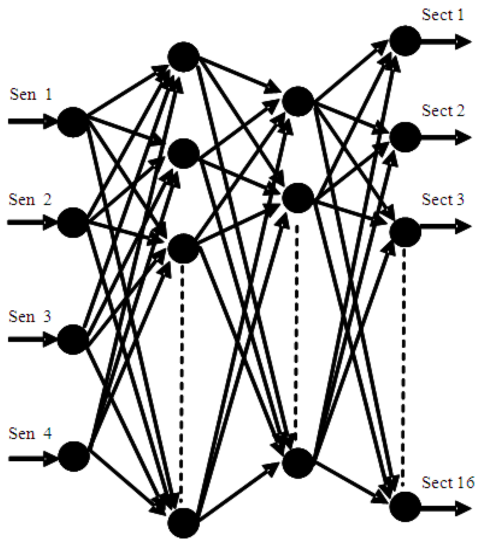


Fig. 6: Neural network structure

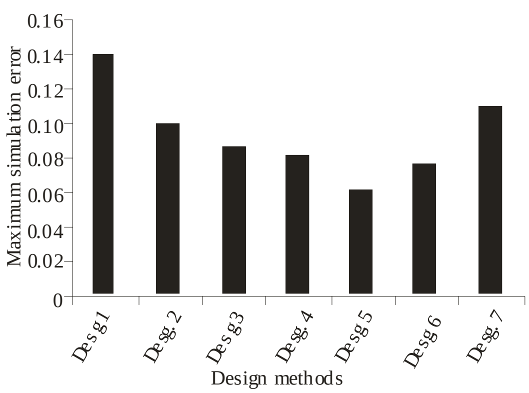


Fig. 7: Neural network evaluation chart, desg.1 (one hidden layer with 5 nodes), desg.2 (one hidden layer with 7 nodes), desg.3(one hidden layer with 11 nodes), desg.4 (two hidden layers, 5 nodes in the first and 3 at the second), desg.5 (two hidden layers, 9 nodes in the first and 5 at the second), desg.6 (two hidden layers, 11 nodes in the first and 5 at the second), desg.7 (two hidden layers, 16 nodes in the first and 10 at the second)

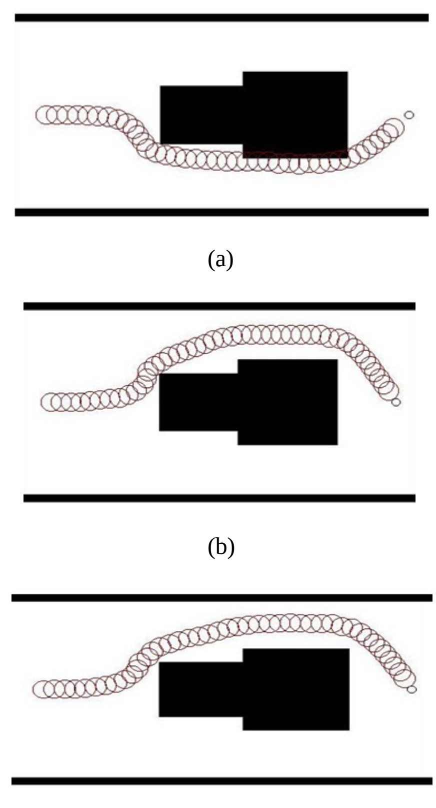
sections in the inference engine. So, one rule at each section will be activated based on the output of the neural network which set in the interval [0.025-1]. The interval [0.025-1] comes from normalize the number of rules in the section based on the maximum number of the rules which is 40. Therefore, the output matrix that has the desired activation rules for each input is set. As a result the size of the output matrix is 16×83521 elements.

Training and learning the neural network: As has been shown in the previous section, the input and output data to train the feed-forward neural network are generated. Then the values of the input matrices are arranged in training vectors in a manner similar to the Jackknife technique (Fukanga, 1990), where 70% of the data were used for the NN training phases and the remaining 30% were used for the NN testing phases.

The topology of the each of the NN consists of four input nodes (Sensor 1, Sensor 2, Sensor 3, Sensor 4), hidden layer and sixteen output node (Sect 1, Sect 2, Sect 3, Sect 4, Sect 5, Sect 6, Sect 7, Sect 8, Sect 9, Sect 10, Sect 11, Sect 12, Sect 13, Sect 14, Sect 15, Sect 16) as shown in Fig. 6. To find the optimum topology (i.e., the optimum number of hidden nodes in the hidden layer), several feed forward NN structures were trained using the training data to find the minimum training error, as shown in Fig. 7. As a result, it is found that the feed forward NN with two hidden layers (9 nodes in the 1st layer and 5 in the 2nd layer) and sixteen nodes for output layer produce the minimum errors.

EVALUATION THE PROPOSED ALGORITHM

The FLC40 is not capable to avoid collision with the edges of the obstacles in all cases due to the low



(a)

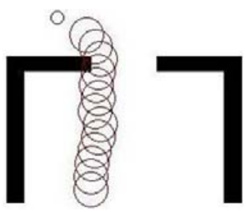
(b)

(c)

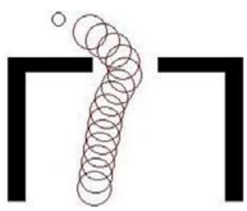
Fig. 8: (A) Result of FLC40; (B) result of FLC625; (C) result of NFOC, simulation results for the three fuzzy controllers

Table 1: Performance evaluation of FLC40, FLC625 and NFOC

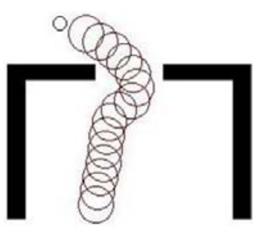
		CPU time		
	Performance	fuzzification processing	inference processing	total CPU time
FLC 40	Slightly colliding with the obstacles	527 µS	132 µS	659 µS
FLC 625	Collision with the obstacles and smoothly reaches the target	1038 µS	1736 µS	2757 µS
NFOC	Collision at all with the obstacles and has a good response	1038 µS	510 µS	1548 µS



(a)



(b)



(c)

Fig. 9: (A) Result of FLC40; (B) result of FLC625; (C) the result of NFOC, simulation results

resolution which is two fuzzy sets. The main problem in the FLC625 presented in reference (Baker, 2002) is the inference block, which consists of a large number of rules that need a long processing time which decrease the response time for the mobile robot as shown in Fig. 8B. To solve this problem of processing time, the inference engine was optimized using neural network. This neural network can process the crisp input value and activated the required fuzzy rules in inference engine in 1a extra short time. Therefore, the processing time for the inference engine is reduced by 70% comparing with traditional inference engine. Comparisons between the three controllers are shown in Fig. 8 and 9.

The main advantage gained by utilizing the NFOC that directly activates the processing activation rules in inference engine. Therefore, there is no need for searching process in rule based in the inference engine. As a result, the processing time is reduced to 70% comparing with the traditional inference engine as shown in Table 1. Practically, simulation-using PC doesn't show the differences in the CPU time for the three controllers since the PC is very fast and the response of the hardware is slow. The CPU time for the three controllers is noticed when using micro controller chip to control the robot motion and download the program to the implemented robot. In the FLC 40, the controller response time will be faster than both controllers, but the performance is limited. On the other hand, the FLC625 worked out well but with low response, which introduced a deficiency in the robot motion (create a dead point in the robot controller). The NFOC increased the response of the whole controller and improves the performance of the robot motion.

CONCLUSION

The performance of the FLC 625 in reference (Baker, 2002) is good and slightly improved the performance of the robot compared to the FLC40 since the robot doesn't touch any obstacle and the robot avoids collision with any obstacles as shown in the above cases. But the inference time is much more than the FLC40. However, the proposed approach that is design based on optimizing the selection of the activation rules in the fuzzy inference engine using the neural network, reduced the processing time and increased the performance for the mobile robot. Therefore, the response time for NFOC on the simulation robot program has shown an excellent reduction with respect to the response time for other fuzzy controllers.

REFERENCES

Aguirre, E. and A. Gonzalez, 2000. Fuzzy behaviors for mobile robot navigation: Design, coordination and fusion. Int. J. Appro. Reas., 25: 255-289.

Baker, A., 2002. An intelligent mobile robot controller using neuro-fuzzy techniques, MSc. Thesis, University of Jordan.

Cao, J., X. Liao and E. Hall, 1999. Reactive navigation for autonomous guided vehicle using the neuro-fuzzy techniques. Eng. Appl. Arti. Int., 6: 320-329.

Callan, R., 1999. The Essence of Neural Networks. 1st Edn., Prentice-Hall, Europe.

Carpenter, G., S. Grossberg, N. Markuzon, H.J. Reynold and D.B. Rosen, 1992. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE T. Neural Net., 3: 698-713.

Fausett, L., 1993. Fundamentals of Neural Networks. 1st Edn., Prentice-Hall, Europe, pp: 461.

Fukanga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press, Missouri, USA. ISBN-10: 0122698517, pp: 591.

Hagras, H. and T. Sobh, 2002. Intelligent learning and control of autonomous robotic agents operating in unstructured environments. IEEE T. Syst. Man Cy. C., 145: 1-2.

Harisha, S.K. and P. Kumar, M. Krishna and S.C. Sharma. 2008. Fuzzy logic reasoning to control mobile robot on pre-defined strip path. World Acad. Sci. Eng. Technol., 42: 642-646.

Medsker, R., 1995. Hybrid Intelligent Systems. Kluwer Academic Publishers, Europe, Retrieved from: http://cs.ioc.ee/yik/lib/22/Medsker1.html.

Ng, K.C. and M.M. Trivedi, 1998. A neuro-fuzzy controller for mobile robot navigation and multirobot convoying. Proceedings of the IEEE Systems, Man and Cybernetics-Part B: Application and Reviews, Dec, 1998, IEEE Xplore Press, La Jolla, CA, 28: pp: 829-840.

Parasuraman, S., B. Shirinzadeh and V. Ganapathy. 2008. Mobile robot navigation using alpha level fuzzy logic system: Experimental investigations. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Oct. 12-15, IEEE Xplore Press, Bandar Sunway, pp: 1878-1884.

Pradhana, S.K., D.R. Parhib and A.K. Pandac, 2009. Fuzzy logic techniques for navigation of several mobile robots. J. Soft Comput., 9: 290-304.

Ross, T.J., 1998. Fuzzy Logic in Engineering Applications. 1st Edn., Mc Graw-Hill Inc, New York.

Singh, M.K., D.R. Parhi, S. Bhowmik and S.K. Kashyap, 2008. Intelligent controller for mobile robot: Fuzzy logic approach. Proceedings of the 12th International Association for Computer Methods and Advances in Geomechanics (IACMAG' 08), pp: 1-6. Retrieved from: http://www.civil.iitb.ac.in/~dns/IACMAG08/pdfs/F20.pdf

Tanaka, K., 1997. An Introduction to Fuzzy Logic for Practical Application. 1st Edn., Spring-Verlag, New York, pp: 138.

Xu, W.L. and S.K. Tso, 1996. Real-time self-reaction of a mobile robot in unstructured environments using fuzzy reasoning. Eng. Appl. Art. Int., 9: 475-485.

Yung, N.H.C. and C. Ye, 1999. An Intelligent mobile vehicle navigator based on fuzzy logic and reinforcement learning. IEEE T. Syst. Man Cy. C., 29: 314-321.

Zadeh, L.A., 1997. The roles of fuzzy logic and soft computing in the conception, design and deployment of intelligent systems. BT Technol., 14: 32-36.