Hybrid Neuro-Fuzzy System for Mobile Robot Reactive Navigation

Ayman A. AbuBaker
Assistance Prof. at Faculty of Information Technology, Applied Science University, Amman-Jordan, a_abubaker@asu.edu.jo.

ABSTRACT
This paper addresses the problem of mobile robot autonomous navigation in a non structured environment. The objective is to make the robot move along a collision free trajectory until it reaches its target. The approach taken here utilizes a hybrid neuro-fuzzy technique where the inference engine of a classical fuzzy system is replaced by a collection of five parallel neural networks in order to reduce computational time for real-time applications. The five neural networks were trained using data sets randomly selected from the original fuzzy decision matrix. Simulation results were conducted to test the performance of the developed system and the results proved that the proposed approach to be practical for real time applications. Finally, the developed neuro-fuzzy controller was tested on a prototype mobile robot which was designed and constructed as part of this research project.

Key Words: Mobile Robot, Neuro-Fuzzy, Obstacle avoidance.

Introduction
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 behavior of human expertise. In general, there are two approaches to the application of fuzzy logic in mobile robot navigation, namely, behavior-based approach [1, 2, 9] and classical fuzzy rule-base approach [3, 4, 6]. The design of fuzzy logic rules is often reliant on heuristic experience and it lacks systematic methodology. Therefore these rules might not be correct and consistent, do not possess a complete domain knowledge, and/or could have a proportion of redundant rules. Furthermore, 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 maybe 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 [5].

Several successful reactive navigation approaches based on neural networks have been suggested in the literature [7, 8, 10]. 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 can not be interpreted into physical knowledge [11]. 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 [12, 14]. 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, neurally adaptive fuzzy inference system, neurally performed FIS, and combinatorial, or hybrid, neuro-fuzzy systems. The neurally 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 [13].

In this paper, a new approach is proposed to design a simple hybrid neuro-fuzzy navigation system. The proposed system has two apparent advantages in structure that simplify and reduce the processing time and improve the performance. First, the if-then rule base is replaced by a set of simple neural networks. However, the second one is the five parallel simple neural networks are utilized to replace the fuzzy inference system acquired by the robot’s sensory system. With such technique, the required time needed to infer the decision for the robot movement is greatly reduced.

2. The Proposed Neuro-Fuzzy Navigation System

The mobile robot is required to explore several paths in a maze, of a pattern of successive combinations 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).

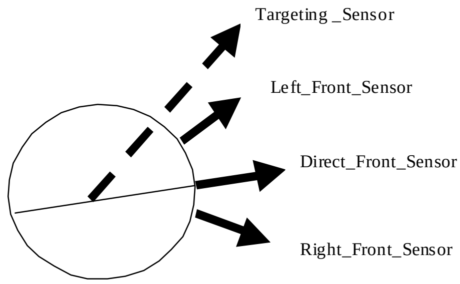


Fig (1) Mobile Robot with Frontal Sensors

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 meters. 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 paper scope.

The four sensors provide the path planning system (in our case a fuzzy logic 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 fuzzy logic controller should pass through three stages, i.e., fuzzification, inference, and defuzzification as shown in Fig (2).

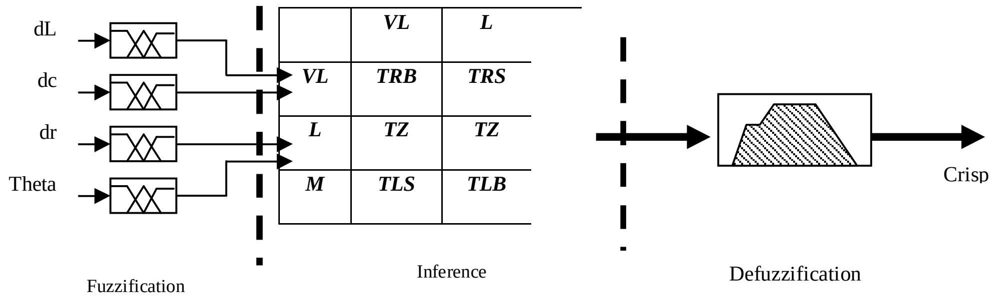


Fig (2) Fuzzy Logic Controller Stages

2.1. Fourty 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 [15]. 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 two to five sets, accordingly, the number of rules was increased to 625 activation rules which will be discussed in the next section.

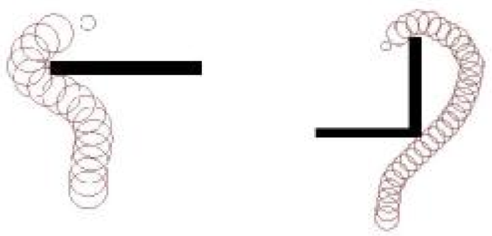
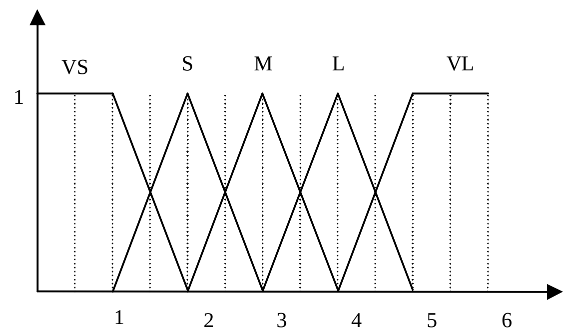


Fig (3) FLC40, simulated motion

2.2. Development of the improved Fuzzy Navigator system

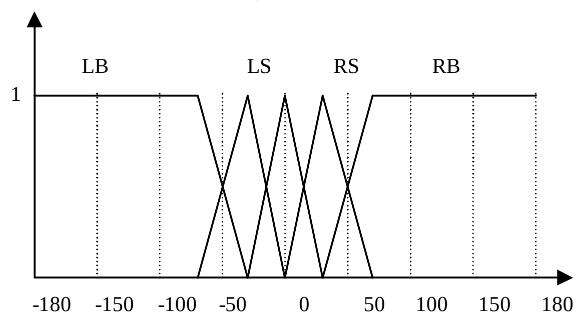
As it has been already noted, that the FLC40 is not capable to avoid collision with the edges of the obstacles in all cases. The main reason behind that failure is the low resolution due to two fuzzy sets, i.e., FAR and NEAR. An improvement to the system can be easily made by increasing the number of fuzzy sets in order to achieve better resolution. In this paper, it is proposed to increase the fuzzy sets to five linguistic labels (VL, L, M, S, VS) as shown in Figure 4(a,b,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 ($5 \times 5 \times 5 \times 5 = 625$ activation rules).



VL: Very Large, L: Large, M: Medium, S: Small,

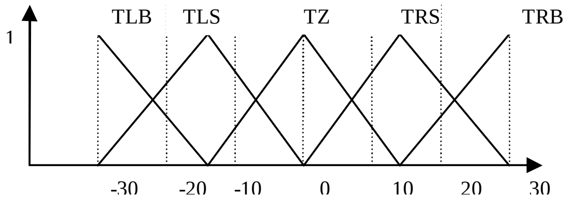
VS: Very Small

(a)



LB: target_ Left Big *LS: target_ Left Small*
,Z: target_Zero
RS: target_ Right Small *RB: target_ Right Big*

(b)



TLB: Turn_ Left _ Big *TLS: Turn_ Left _ Small*
TZ: Turn_ Zero
TRS: Turn_ Right _ Small *TRB: Turn_ Right _ Big*

(c)

Figure 4: a) Distance b) target orientation c) Turning Angle Membership function

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

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

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.

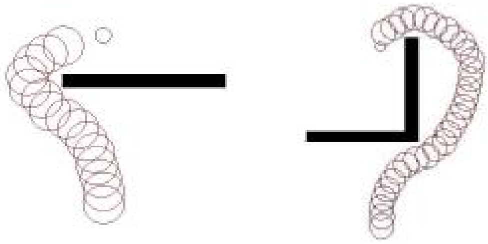


Fig (5) Improvement FLC625 success cases

2.3. Development of the Neuro-Fuzzy Navigation System

The main problem in the fuzzy logic controller is the inference block, which consists of a large number of rules that need a long processing time. To solve this problem of processing time, the inference engine was replaced with a neural network. The system is investigated by considering the results of the integration between both systems (Fuzzy logic and neural networks) as shown in Fig 6.

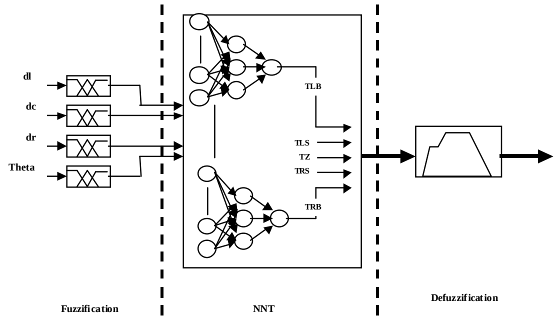


Fig (6) Hybrid Neuro- Fuzzy Architecture

The outcome has been efficient and accurate but it requires training, which was introduced for that system. To perform the training process, the sample turns out to be very large (83521×20) and the system faces problems. To overcome that huge sample, the NN was structured as five parallel networks where each one of this network has 20 fuzzy input nodes and one fuzzy output node as shown in Fig (7). This new structure has improved the performance and the response time.

The main idea in this neuro-fuzzy system is the replacement of inference engine by neural networks where it has a fuzzy inputs and a fuzzy output.

The performance of the Hybrid neuro- fuzzy controller is the same as the improved fuzzy logic controller. The neural network in this controller is trained to do the same action as the inference engine in the improved fuzzy logic controller as shown Figure 8 a, b.

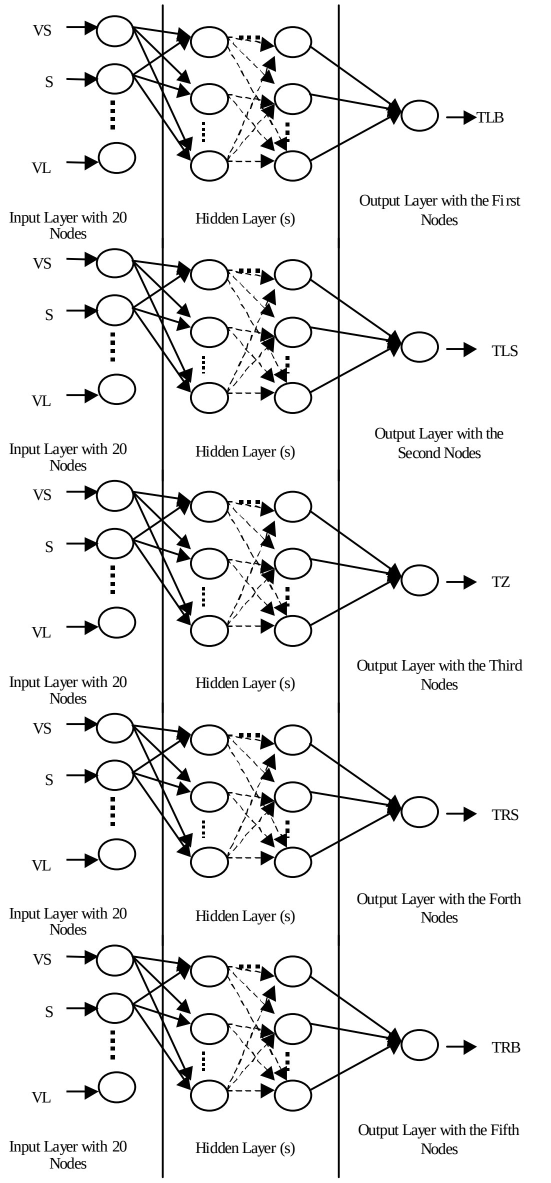


Fig (7) Five Neural Network with Parallel Inputs and Five Separated Outputs

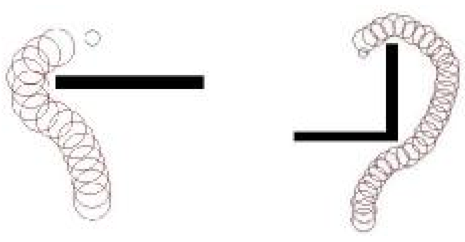


Fig (8) NFC Success Cases

The main advantage gained by utilizing the hybrid neuro-fuzzy controller is the reduction in the inference time from 1736 µS to 456 µS which increases the response of the controller and improves the performance of the robot. 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 FLC40, 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 neural network which programmed in five chips, the data of the main micro-controller entered to the five parallel NNT and this increased the response of the whole controller and improves the performance of the robot motion.

However, simulation experiments were conducted to test the performance of the developed controller and the results proved that the approach is suitable to be used in practical design for real time applications. For example, the inference CPU time of fourty rules fuzzy navigator is measured to be 132 µS with slightly or severely colliding with obstacles. To avoid collision with obstacles, the inference engine rules were increased to 625. This improvement increased the inference CPU time by 13 times. A model of neuro-fuzzy controller with single node achieved better performance – no collision with obstacles- with inference CPU time reduction by 38% of 625 inference engine time.

3. Conclusion

The performance of the FLC625 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 based on using neuro-fuzzy system instead of the inference engine is reduced the processing time and increased the performance. The response of the implemented robot has shown an excellent reduction with respect to the response time.

Table 1: Performance Evaluation of FLC40, FLC625, and NFC.

	Performance	CPU Time		
		Fuzzification Processing	Inference Processing	Total CPU Time
FLC 40	Slightly or severely colliding with the obstacles	527 µS	132 µS	659 µS
FLC 625	Avoid collision with the obstacles and smoothly reaches the target	1038 µS	1736 µS	2757 µS
NFC with Single Node	No collision at all with the obstacles & has a good response	1038 µS	456 µS	1494 µS
NFC with 5 Nodes	No collision at all with the obstacles & has a good response	1038 µS	2230 µS	3268 µS

Reference:

[1] Aguirre E. and González A. 2000. Fuzzy behaviors for mobile robot navigation: design, coordination and fusion. *International Journal of Approximate Reasoning*, 25(3): 255-289.

[2] Lee C. C. 1990. Fuzzy logic in control systems. Fuzzy logic controller- part 1 and part *IEEE Trans. on systems, Man and Cybernetics*, 20 (2): 404-435.

[3] Proychev T. and Petrov M. 1996. Fuzzy Navigator of Mobile Robots in Unstructured Environment. *IEEE Transaction on Systems, Man, and Cybernetics-Part B: Application and Reviews*, 26(5).

[4] Ross T.J. 1998. *Fuzzy Logic in Engineering Applications.* Mc Graw-Hill Inc, New York.

[5] Tanaka K. 1996. *An Introduction to Fuzzy Logic for Practical Application.* 1st Edition. Spring-Verlag, New York.

[6] Zadeh L. A. 1996.The roles of fuzzy logic and soft computing in the conception, design and deployment of intelligent systems. *BT Technology*, 14(4): 32-36.

[7] Callan R. 1999. *The Essence of Neural Networks*. Prentice-Hall, Europe.

[8] Carpenter G., Grossberg S., Markuzon N., Reynold H. J. and Rosen D. B. 1992. Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analogue Multi-dimensional Maps. *IEEE Tran on Neural Networks*, 3(5): 698-713.

[9] Saffiotti A. 1997. Fuzzy Logic in Autonomous Robotics: behavior coordination. *IEEE International Conference of Fuzzy Systems*: 573-578.

[10] Fausett L. 1994. *Fundamentals of Neural Networks*. Prentice-Hall, Europe.

[11] Hagras H. and Sobh T. 1998. Intelligent Learning and Control of Autonomous Robotic Agents Operating in Unstructured Environments. *IEEE Transaction on Systems, Man, and Cybernetics-Part C: Application and Reviews*, 26(4).

[12] Cao J., Liao X. and Hall E. 2000. Reactive Navigation for Autonomous Guided Vehicle Using the Neuro-fuzzy Techniques. *Engineering Application of artificial intelligent*, 6(4): 320- 329.

[13] Medsker A. R. 1995. *Hybrid Intelligent Systems.* Kluwer Academic Publishers, Europe.

[14] Yung N. H. C. and Ye C. 1999. An Intelligent Mobile Vehicle Navigator Based on Fuzzy Logic and Reinforcement Learning. *IEEE Transaction on Systems, Man, and Cybernetics-Part C: Application and Reviews*, 29(2).

[15] Xu W.L., Tso S.K. 1996. Real time Self–Reaction of a Mobile Robot in Unstructured Environment Using Fuzzy Reasoning. *Engineering Application of artificial intelligent*, 9(5): 475- 485.