Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

An Embedded Interval Type-2 Neuro-Fuzzy Controller for Mobile Robot Navigation

Siti Nurmaini [1] , Siti Zaiton [2] , Dayang Norhayati [2]

[1] Faculty of Computer Science, University of Sriwijaya, Indonesia

[2] Faculty of Computer Science and Information System,UTM, Skudai, Johor Bahru

siti_nurmaini@unsri.ac.id, sitizaiton@utm.my, dayang@utm.my

***Abstract*— This paper describes intelligent navigation using an embedded interval type-2 neuro–fuzzy controller. Weightless neural network (WNNs) strategy is used because fast learning, easy hardware implementation and well suited to microcontroller-based-real-time systems. The WNNs utilizes previous sensor data and analyzes the situation of the current environment and classifies geometric feature such as U-shape, corridor and left or right corner. The behavior of mobile robot is implemented by means of interval type-2 fuzzy control rules can be generated directly from the WNNs classifier. This functionality is demonstrated on a mobile robot using modular platform and containing several microcontrollers implies the implementation of a robust architecture. The proposed architecture implemented using low cost range sensor and low cost microprocessor. The experiment results show, using that technique the source code is efficient. The mobile robot can recognize the current environment and to be able successfully avoid obstacle in real time and achieve smother motion compare than logic function and fuzzy type-1 controller.**

Keywords: Interval type-2 fuzzy; WNNs classifier, embedded controller; mobile robot; navigation

I. INTRODUCTION

The utility of robotic model-based reasoning approach for the design of intelligent robots is limited due to uncertainties inherent to unstructured environments, unreliable and incomplete perceptual information and imprecise actuators [7]. In many applications the robot's environment changes with time in a way that is not predictable by the designer in advance. In addition, the knowledge about the environment is often imprecise and incomplete due to the limited perceptual quality of sensors.

Fuzzy system employ a mode of approximate reasoning that makes them a suitable tool to implement a robust robot behavior tolerating noisy and unreliable sensor information. The fuzzy rules describe the relation between the external and internal states of the robot and the set of possible actions [15]. To date all the FLC implementation in robot control are based on the traditional type-1 FLC. The most common way is to construct the FLC by eliciting the fuzzy rules and the membership functions based on expert knowledge or through the observation of the actions of a human operator controlling the mobile robot [15][16].

However, the type-1 FLCs have the common problem that they cannot fully handle or accommodate for the linguistic and numerical uncertainties associated with changing and dynamic environment because they use precise type-1 fuzzy sets [7]. Recently, Mendel and Karnik [9] have developed a type-2 fuzzy logic system, this method can handles imprecise and uncertainty data to produce complex decision outcomes, it fuzzy sets let us model and minimized the effects of uncertainties.

Although, many fuzzy logic technique for mobile robot navigation have been proposed in the literature, a few using a type-2 fuzzy logic. Hagras et al, [7] present a hierarchical type-2 fuzzy logic control for mobile robot navigation in outdoor and indoor unstructured environment. That paper implements the basic navigation behaviors and coordination between these behaviors to make control decision. Baklouti et al, [19], they deals with the design of an Interval Type-2 TSK FLC for the navigation of mobile robots in unknown and dynamic environments. The purpose of this controller is to perform the navigation in environments using polygonal robots from an initial point to a designed goal. P. Phokharatul [20], they present behavior base control for mobile robot using type-2 fuzzy logic system. The method used in the construction of algorithm representing the basic behaviors of mobile robot.

Recently, intelligent control approach has been done on applications of fuzzy neural networks (FNN), which combine the capability of fuzzy reasoning to handle uncertain information and the capability of artificial neural networks to learn from processes. Currently, there were only few works to analyze and simulate the interval type-2 FNN to construct advanced controllers based on the back propagation algorithm [11]-[13]. However, conventional neural networks often require powerful computer systems to implement in a reasonable time-frame [6], due to the large size of resource because the training the generalized data required both the forward propagation phase and backward propagation phase. So, that technique can be problematic for implemented in

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

embedded application using a low cost and simple microcontroller with modest capabilities in terms of both speed and memory.

WNNs is a good choice for use with simple microprocessor because in contrast to the weighted neural models, there are several one-shot learning algorithms, for WNNs where training takes only one epoch. Several works done by others indicates this approach may be successful. Mitchell et al [8] has shown that a simple Hopfield network can be used to control an insect robot's movement, which can move forwards, backwards, or move around an obstacle, by using only a Z80-based microcontroller system. Zhou et al [10], a simple perceptron network was developed to control a fire-fighting robot, the neural network was used to detect obstacle, find a fire, and then the robot moved to extinguish the fire. Yao et al [6] a RAM base neural networks is being developed for a mobile robot controlled by a simple microprocessor system, the RAM-based neural network allows the robot to detect and avoid obstacles in real time. In [3], also referred to as weightless network, when comparing RAM-based neuron with multi-layer network, it has significant advantages [4] such as: fast learning and easy hardware implementation, because a one-shot learning. WNNs strategy is employed in RAM-base neurons which require only one pass through the training set and only one access to memory. The process training very quickly, this is especially well suited to microcontroller-based-real-time systems because it requires very simple computing capabilities.

The purpose of this paper is to development combination a type-2 fuzzy neural network for detection local obstacle and recognition an unknown environment for making navigation decision. Our earlier works using interval fuzzy type-2 controller on mobile robot explained in [17], and environmental recognition base RAM network in [18].

This paper is organized as follow: Section 2 presents the problem description system platform. Section 3 introduces the embedded controller using interval fuzzy type-2 and neural network. Section 4 provides experimental result of the navigation using the controller described in section 3. Finally, Section 5 presents the conclusions.

II. THE SYSTEM PLATFORM

The robot built shown in Fig. 1, it has a cylindrical shape, measuring 20 cm in diameter, and 17 cm height. This robot has two wheels in the both side and one free wheel at the back side. The two parallel wheels are driven by DC motors. We use two microcontroller AT89x55 for central controller, attached with ROM 20 Kbytes, RAM 256 bytes and clock 24.3 MHz, operated in 0.5 micro-second for each process. A general purpose high performance controller for decision making is connected to the parallel bus. In this work we utilized eight ultrasonic sensor located at the front, left and right side of the robot. For process signal in sensor module, eight PIC16F84 chips are use, with ROM 1 Kbytes and operated in 1 micro-second. The sensor Attached to the mobile robot, each separated at 30^o along the circumference. Furthermore, for real time control applications, PIC18F2550 is used exclusively to generate the PWM signals and to run the two dc motors, attached to each motor is an optical encoder which used for distance and velocity calculation. Several peripheral features are available including: timer/counter, two PWM modules, two serial ports, 8 bit wide parallel port and 8-chanel high peed 8-bit a/d converter.

The architecture of mobile robot in Fig.1 is a modular system and containing several microcontrollers implies the implementation of a robust communication mechanism between modules. For this platform, the architecture can conceptually be seen as a central control module interfacing all other functional modules, which either supply or demand data required for autonomous processing as shown in Fig 2. The central control module, it implements the algorithms of the navigation control it consist interval type-2 fuzzy logic controller and weightless neural network classifier. The sensor module is configured it controls the ultrasonic pulses, calculates the sensor readings, and selects which sensors are active according their position in the robot periphery. The motor drive module is configured the speed levels of the robot. The neural network classifies the environment using information from sensor module into geometric features such as U-shape, corridor and left or right corner and utilizes previous sensor data to analyze the situation. The fuzzy logic controller uses the result from neural network for making navigation decision.

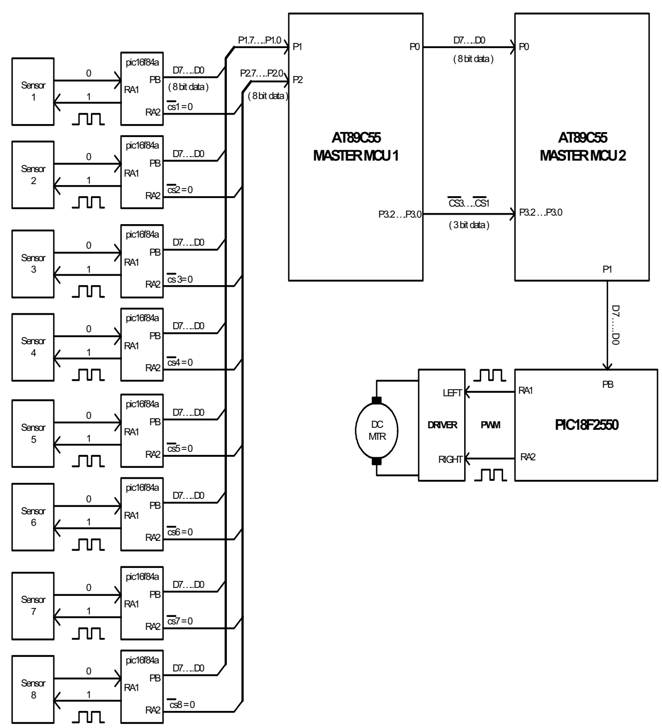

Figure 1. The robot system

III. THE EMBEDDED INTERVAL TYPE-2 NEURO-FUZZY CONTROLLER

A. Interval Fuzzy type-2 Controller Design

Interval type-2 FLCs that use interval type-2 fuzzy sets have been proven to be able to model and handle the uncertainties to give a good performance [7], [12], [22], [23]. However, the interval type-2 FLC involves a computational overhead associated with the type-reduced fuzzy sets using Karnik-Mendel procedure [23], [14] which can reduce the robustness performance of the type-2 FLC especially when operating on embedded platforms. Wu and Mendel [15] introduced a method to approximate the type-reduced set by the inner and outer bound sets, thus avoiding the use of the iterative Karnik-Mendel procedure.

An interval type-2 fuzzy set $\widetilde{F}$ can be written as follow

$$\widetilde{F} = \int_{x\in X}\left[\int_{u\in[\underline{\mu}_{\widetilde{F}}(x),\overline{\mu}_{\widetilde{F}}(x)]} 1/u\right]/x \tag{1}$$

where $\underline{\mu}_{\widetilde{F}}(x), \overline{\mu}_{\widetilde{F}}(x)$ represent the upper and lower MFs respectively. We extended the type-1 fuzzy system controller to an interval type-2 fuzzy system, by adding uncertainties in both antecedent and consequent part of each rule. So we spread the MFs values of the antecedent part by ± A%, and the consequent part by ± C%. For each input *k* and rule *i*, is represented by triangular membership function with uncertain mean. The upper and lower MFs for this interval type-2 fuzzy set can be written in equation (2) and (3) respectively,

$$\overline{f}_A(x) = \begin{cases} 0 & x\langle l_1 \\ \dfrac{x-l_1}{p_1-l_1} & l_1 \le x\langle p_1 \\ 1 & p_1 \le x \le p_2 \\ \dfrac{r_2-x}{r_2-p_2} & p_2\langle x \le r_2 \\ 0 & x\rangle r_2 \end{cases} \tag{2}$$

$$\underline{f}_A(x) = \begin{cases} 0 & x\langle l_2 \\ \dfrac{x-l_2}{p_2-l_2} & x \le \dfrac{r_1(p_2-l_2)+l_2(r_1-p_1)}{(p_2-l_2)+(r_1-p_1)} \\ \dfrac{r_2-x}{r_2-p_2} & x\rangle \dfrac{r_1(p_2-l_2)+l_2(r_1-p_1)}{(p_2-l_2)+(r_1-p_1)} \\ 0 & x\rangle l_2 \end{cases} \tag{3}$$

In this research we use the type-2 singleton fuzzifier, the upper $\overline{\mu}_{\widetilde{F}_k^i} x(k)$ and lower $\underline{\mu}_{\widetilde{F}_k^i} x(k)$ membership values are calculated using Equations (2) and (3) respectively. The firing strength f^i of the i^{th} rule is an interval type-1 set determined by its left most point $\underline{f}^i$ and its right most point $\overline{f}_i$ which are calculated as follows [21]:

$$\underline{f}^i = \underline{\mu}_{\widetilde{F}_1^i}(x_1) * \ldots\ldots\ldots * \underline{\mu}_{\widetilde{F}_n^i}(x_n) = \prod\nolimits_{j=1}^{n} \underline{\mu}_{\widetilde{F}_j^i}(x_j) \tag{4}$$

$$\overline{f}^i = \overline{\mu}_{\widetilde{F}_1^i}(x_1) * \ldots\ldots\ldots * \overline{\mu}_{\widetilde{F}_n^i}(x_n) = \prod\nolimits_{j=1}^{n} \overline{\mu}_{\widetilde{F}_j^i}(x_j) \tag{5}$$

The inference engine computes the degree of firing of each rule by using the meet operation under the product t-norm between the antecedent membership grades of each rule. The defuzzification layer approximates the type-reduced set using the Wu-Mendel Uncertainty Bounds. It provides mathematical formulas for the inner and outer bound sets which can be used to approximate the type-reduced set [21]. Equations (6) and (7) define the inner bound set while (8) and (9) define the outer bound set [21].

$$\overline{y}_l = \min\left\{\frac{\sum_{i=1}^{M}\underline{f}^i w_l^i}{\sum_{i=1}^{M}\underline{f}^i}, \frac{\sum_{i=1}^{M}\overline{f}^i w_l^i}{\sum_{i=1}^{M}\overline{f}^i}\right\} \tag{6}$$

$$\underline{y}_r = \max\left\{\frac{\sum_{i=1}^{M}\underline{f}^i w_r^i}{\sum_{i=1}^{M}\underline{f}^i}, \frac{\sum_{i=1}^{M}\overline{f}^i w_r^i}{\sum_{i=1}^{M}\overline{f}^i}\right\} \tag{7}$$

$$\underline{y}_l = \overline{y}_l - \left[\frac{\sum_{i=1}^{M}(\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^{M}\overline{f}^i \sum_{i=1}^{M}\underline{f}^i} * \frac{\sum_{i=1}^{M}\underline{f}^i(w_l^i - w_l^1)\sum_{i=1}^{M}\overline{f}^i(w_l^M - w_l^i)}{\sum_{i=1}^{M}\underline{f}^i(w_l^i - w_l^1) + \sum_{i=1}^{M}\overline{f}^i(w_l^M - w_l^i)}\right] \tag{8}$$

$$\overline{y}_r = \underline{y}_r + \left[\frac{\sum_{i=1}^{M}(\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^{M}\overline{f}^i \sum_{i=1}^{M}\underline{f}^i} * \frac{\sum_{i=1}^{M}\underline{f}^i(w_r^i - w_r^1)\sum_{i=1}^{M}\overline{f}^i(w_r^M - w_r^i)}{\sum_{i=1}^{M}\underline{f}^i(w_r^i - w_r^1) + \sum_{i=1}^{M}\overline{f}^i(w_r^M - w_r^i)}\right] \tag{9}$$

The crisp outputs in defuzzification layer can be computed as follows [21]:

$$y(x) = \frac{\dfrac{\underline{y}_l + \overline{y}_l}{2} + \dfrac{\underline{y}_r + \overline{y}_r}{2}}{2} \tag{10}$$

The robot behaviors was divided into four behaviors namely obstacle avoidance, left wall following, right wall following, and emergency condition. Linguistic for obstacle distance is represented by fuzzy set near, medium, far and for wall distance is represented by near and far in Fig 2a, 2b. Output fuzzy set is represented by steering angle with linguistic are small, medium, big as shown in Fig.2c, and speed of the motor are slow, medium, fast in Fig. 2d.

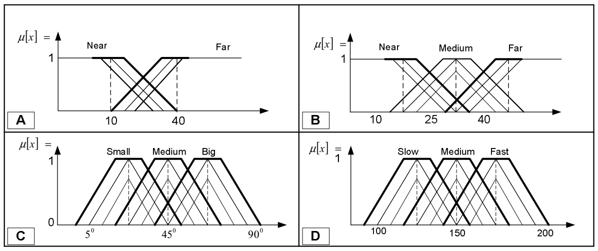

Figure 2. Membership function a). wall distance b). obstacle distance c). steering angle d) speed of the motor

When the robot is very close to the target, the attractive force between the robot and the target causes the robot seeking towards the target. Similarly when the robot is very close to an obstacle, because of repulsive force developed between the robot and the obstacle the robot must change its speed and heading angle to avoid the obstacle. Some of fuzzy rules used for four behaviors for change the heading angle are listed in table 1. Some rules mentioned in table 1 cater for extreme conditions when the obstacles have to be avoiding as quickly a possible. All rules in that table have been obtained heuristically using common sense.

TABLE 1. RULE BASE FOR FOUR BEHAVIOR

No	Obstacle Avoidance			Action
	Left	Front	Right	
1	Far	Far	Far	Forward
2	Far	Far	Medium	Turn left
3	Far	Far	Near	Turn left
4	Far	Medium	Far	Turn right
5	Far	Medium	Medium	Turn right
6	Far	Medium	Near	Turn left
7	Far	Near	Far	Turn right
8	Far	Near	Medium	Turn left
9	Far	Near	Near	Turn left
10	Medium	Far	Far	Turn right
11	Medium	Far	Medium	Forward
12	Medium	Far	Near	Turn left
13	Medium	Medium	Far	Turn right
14	Medium	Medium	Medium	Forward
15	Medium	Medium	Near	Turn left
16	Medium	Near	Far	Turn right
17	Medium	Near	Medium	Turn left
18	Medium	Near	Near	Turn left
19	Near	Far	Far	Turn right
20	Near	Far	Medium	Turn right
21	Near	Far	Near	Backward
22	Near	Medium	Far	Turn right
23	Near	Medium	Medium	Turn right
24	Near	Medium	Near	Backward
25	Near	Near	Far	Turn right
26	Near	Near	Medium	Turn right
27	Near	Near	Near	Backward
	Right Wall Following			**Action**
	Right 1	**Right 2**		
28	Far	Far	-	Forward
29	Far	Near	-	Turn left
30	Near	Far	-	Turn left
31	Near	Near	-	Turn left
	Left Wall Following			**Action**
	Left 1	**Left 2**		
32	Far	Far	-	Forward
33	Far	Near	-	Turn right
34	Near	Far	-	Turn right
35	Near	Near	-	Turn right

B. WNNs Classifier Design

RAM is the most cost effective digital component and has some properties in common with a McCulloch and Pitts (MCP) neuron. The network can be trained by writing to memory and the trained data can then be recalled (read). The basic element of that classifier is a bit-addressable RAM discriminator that is formed by a layer of k N-input RAMs, each one storing 2^{N} one bit words, see Figure 3. Each of these k RAMs is addressable by an N-tuple, which is selected randomly from an array of binary values forming the input to the network. Although the input mapping is chosen at random, such a mapping is often a fixed parameter of the network. At the discriminator's output, an adder sums the outputs of the RAMs, producing what is called the discriminator's response r. Before training takes place, all the k 2^{N} one-bit words are set to zero, and training consists in their modification.

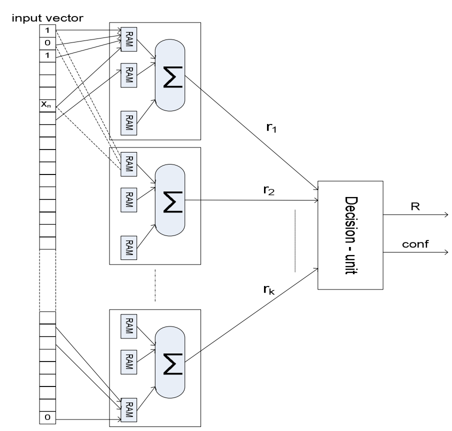

Figure 3. RAM base network

The weightless neural network is called a multi-ram discriminator as shown in Fig. 3. In this way the input pattern of the discriminator is distributed over the memory of several RAM-nodes, instead of being stored in only one location. The generalization is obtained from a summing device to sum up the outputs of all the RAM-nodes of that discriminator and divide it by the number of RAM nodes *K*. The gives fraction of RAM nodes that generated a '1' when input vector *X* was given to the discriminator:

$$r_j = \frac{1}{K}\sum_{i=1}^{k} f_{ji}(x) \tag{11}$$

where, r_j is output of the j_{th} discriminator, f_{ji} is output of RAM node i belonging to the j_{th} discriminator, K is number of RAM nodes and x_i is the i_{th} element of X.

In this way the result r of the discriminator can be seen as the outcome of a membership function, it is a number between 0 and 1 which states how much pattern X resembles the pattern used in training set for discriminator. The set of feature vectors generated by the WNNs for a given training set will be used as input data to the fuzzy rule-based system. The normalized response of class j discriminator x_j is defined by:

$$x_j = \frac{r_j}{K} \tag{12}$$

The value of x_j can be used as a measure of the similarity of the input pattern to the j_{th} class training patterns. The normalized response vector generated by all discriminators for an input pattern is given by:

$$\mathbf{x} = [\,x_1, x_2, \ldots\ldots, x_N\,]\,, \quad 0 \le x_j \le 1 \tag{13}$$

This response vector can be regarded as a feature vector that measures the similarity of an input pattern to all classes. During the recalling, the system presents the input vector to all trained discriminator. The results r_j of the discriminators j_{th} are passed to a calculation unit. This unit determines which discriminator gives the highest output. This is the class to which the input pattern is classified. It also calculates a measure of relative confidence C:

$$C = \frac{r_{highest} - r_{secondhighest}}{r_{highest}} \tag{14}$$

In this research the modular WNNs is designed to identify the current environment and enhance the generalization, the 8 bit data from eight ultrasonic sensors is used to determine the obstacle. The combination of them appearance in the seven directions makes up different input pattern, the structure is shown in Fig. 4. The network used eight neurons for identifies where obstacle may lie in seven directions. Using only seven output commands means that the neural network needs eight bits to encode them and seven classes of discriminators to work. Its configuration has seven groups of eight neurons (*m*=8 and *n*=7) each neuron has eight bits of memory to store its contents. The neural network has 56 neurons and the winner-takes-all chooses the command for the winning class.

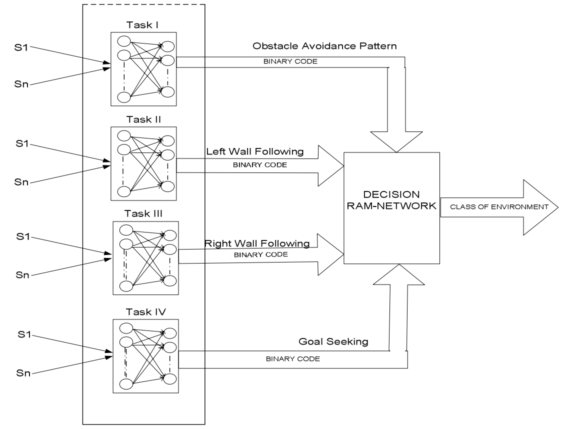

Figure 4. Structure of modular WNNs

The output layers have seven neurons, there are seven position of obstacle in the robot environment, in the front, right or left side of the robot.

C. Environmental Classification

General indoor environment that the robot navigates can be grouped into nine situations as shown in Fig. 5. The number of neurons in the output layer represents the number of subspaces to be classified with nine different geometric features starting with no obstacle detected at all and ending with U-shape. The calculation of this value is based on the distance from an obstacle to the robot. Here we use patterns with a single far, single medium, or single near obstacle to train the neural network. The threshold values for training are, 00100010 (30 cm) that mean far, 00010111 (20 cm) that mean medium, 00101111 (10 cm) that mean near and 01001011 (75 cm) indicating that no obstacle is detected.

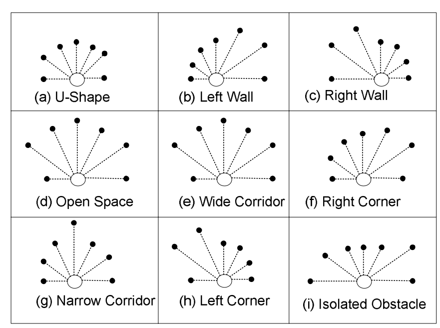

Figure 5. Choosing the training value

At a time the obstacle is place in different directions and in difference distance. The neural network then is taught that the obstacle at left side, right side or forward. Using this technique, the value distinguishing distance an obstacle has to be obtained first. In the evaluation phase, the RAM-base network work by generating all the possible input combination. The number of possible combination of the eight sensors is 2^8 = 256 combination. Then, each output for all input possibilities was written to a lookup table, representing the neuron combination and the signal from sensors is directly connected to the address lines of the memory.

D. Learning/Recalling Strategy

The learning of WNNs has no weight matrix, it directly changing the neuron contents in the look-up tables [5]. The procedures of learning/recalling process takes places by writing/reading into corresponding look up table entries. Initially all the contents are set to 0 and the learning of an input pattern is through the writing of the value 1 in the address content of the RAM neuron. During the recalling phase an input pattern is clamped to the network and all the neurons produce an output. The algorithm used to train is as follow.

- Initialize all location of the memory to a random binary [1,0] values and created the discriminator
- Defined recognition interval by the parameter $0 \leq r_{\min} \leq r_{\max} \leq 1$, $r_{\max}$ being the maximum recognition
- Select an input pattern of the environment from sensor array
- Access the RAM and generate an output,
- If the value at the output of the network is occurred and correct, r set to 1,
- If the value at the output of the network is occurred and incorrect, r set to 0
- Check value of recognition interval r, if $r_h \leq r_{\min}$, a new discriminator is created
- Check value of recognition interval r, if $r_{\min} \leq r_h \leq r_{\max}$, it is assumed that the pattern to be performed is probabilistic selected
- Check value of recognition interval r, if $r_h \leq r_{\max}$, it is assumed that the pattern is already well represented the winning class of discriminator
- For all nodes, if r =1 then set input pattern to be learned, if r = 0, then clear all the nodes and reenter input pattern.
- Go to access the RAM

IV. EXPERIMENTAL RESULT

Experiment is conducted to demonstrate the ability of a mobile robot to react to various unknown environment. Using winner take all decision, achieved 98% classification for geometric feature such as U-shape, plane, corridor and left or right corner. How ever the poorest result was if the robot closes the object, where the scanning sensory sector of the robot was quite high and some noise has still interfered in echo signal, The result is based on the environment classification by the WNNs can show in Table 2.

In this experiment all sensors set to work at medium range and the velocity of the motors was set to the minimum speed. The upper bound and lower bound of the robot steering angle and distance of the obstacle was used 30 % and 10 % respectively.

TABLE 2. ENVIRONMENT RECOGNITION

Actual Place	Distance (cm)	Reference (hex)	Result (hex)
Convex (90^o)	20	12h	12h
	30	0bh	0bh
	40	05h	0dh
Concave (270^o)	10	06h	00h
	20	0eh	00h
	30	00h	00h
Plane (180^o)	10	03h	04h
	20	0ch	0ch
	30	15h	15h
	40	1eh	1eh
Left-corner (180^o)	10	03h	07h
	20	0ch	0ch
	30	15h	15h
	40	1eh	1eh
Right-corner (180^o)	10	03h	07h
	20	0ch	0ch
	30	15h	15h
	40	1eh	1eh
Corridor (0^o)	10	03h	05h
	20	0ch	0ch
	30	15h	15h
	40	1eh	1eh
U-shape (180^o)	10	03h	03h
	20	0ch	0ch
	30	15h	15h
	40	1eh	1eh

From the experiment controller classifies immediate environment by recognizing the pattern that were encountered during training. The WNNs great simplicity ad their implementation as elementary logic functions is responsible for their greater performance.

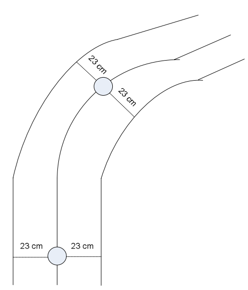

Figure 6. Detecting a wall and corridor

From the experiment, the mobile robot learned to go forward in the open field there was no obstacle. When an obstacle was put in its way, it was able to find the obstacle and select an appropriate action. The base address 30H in hexadecimal is the address in memory of the first byte of the 256 locations of the memory. The output type-2 fuzzy logic controller chooses appropriate action base on the environment classification and control the robot for several cases. For example, the mobile robot can control a distance to the wall of corridor and follow the wall on the left or the right side, the distance between robots and wall about 23 cm base on Fig. 6. The mobile robot moves forward follow the corridor smother better than fuzzy type-1 and logic function controller as shown in Fig. 7.

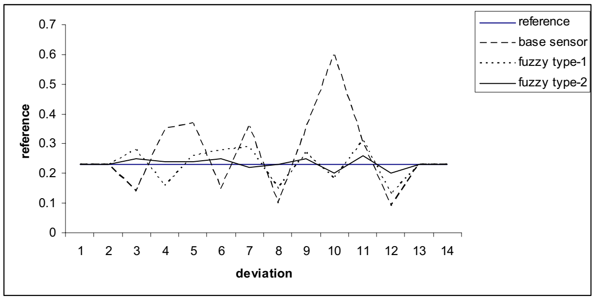

Figure. 7. Wall following behavior for 10 m track

V. CONCLUSIONS

The result has shown, although some noise has still interfered in echo signal the proposed architecture achieved 94 % classification for geometric feature such as, U-shape, corridor and left or right corner using ultrasonic sensors. Interval type-2 fuzzy and WNNs classifier can minimize the execution time because the program of classifier converted in to a look-up table and implemented using modular WNNs. The algorithm was implemented with very modest microcontroller system and source code using the assembler and C code only 750 bytes for WNNs and 22 Kbytes for interval type-2. From the experiment the robot was able to detect an environment and avoid obstacles in real time. The performance of the proposed architecture have yielded promising results that indicate the mobile robot can recognize the current environment, achieved robust performance and was able to successfully perform the navigational tasks for wall following and obstacle avoidance compare logic function, T1FL controller. A sophisticated robot is being developed to improve its ability to detect objects and to show the ability of an embedded interval fuzzy type-2 and neural network classifier implemented on a simple microprocessor platform to perform the difficult task of achieving a destination objective in an unknown environment.

ACKNOWLEDGMENTS

This work was supported by Faculty of Computer Science, Sriwijaya University, Indonesia and Faculty of Computer Science and Information System, UTM, Johor Bahru.

REFERENCES

[1] J.J. Hopfield, "Neural Networks and physical systems with emergent collective properties" ,Proc. Nat. Acad. Sci., USA 79, 2554-8, 1982.

[2] Botelho, S. C., Simoes, E. D. V., Uebel, L. F., and Barone, D. A. C., High Speed Neural Control for Robot Navigation. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Beijing, China, pp. 421-429, 1996.

[3] J, Austin, “A Review of RAM-Based Neural Networks”. Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems, pp. 58-66, 1994.

[4] Simoes, E. D. V., Uebel, L. F., and Barone, D. A. C., Hardware Implementation of RAM Neural Networks. In Pattern Recognition Letters, n. 17, pp. 421-429, 1996.

[5] L, Teresa., Carvalho, A., Brage, A., and Souto, M., Weightless Neural Models: a Review of Current and Past Works. In Neural Computing Surveys, v. 2, pp. 41-61, 1999.

[6] Q. Yao, D. Beetner, D.C. Wunsch, and B. Osterloh., “A RAM-Based Neural Network for Collision Avoidance in a Mobile Robot”, IEEE, 2003

[7] H. Hagras, "A Hierarchical type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots," IEEE Transactions On Fuzzy Systems, vol. 12 No. 4, pp. 524-539, August 2004.

[8] R.J. Mitchell, and D.A. Keating, “ Neural Network Control of Simple Mobile Robot”, Internal report, Departement of Cybernetics, University of Reading, UK, April 1994

[9] J. Mendel and R. John, “Type-2 fuzzy sets made simple,” IEEE Trans. On Fuzzy System, Vol. 10, PP 117-127, April, 2002.

[10] Y. Zhou, D. Wilkins, R. P. Cook, “Neural Network for a fire-fighting robot”, University of Mississippi

[11] C.Wang, C.Cheng, and T.Lee, "Dynamical Optimal Training for Interval Type-2 Fuzzy Neural Network (T2FNN)," IEEE Trans. On Systems, Man and Cybernetics Part B: Cybernetics, vol. 34, no. 3, pp. 1462-1477, June 2004

[12] C. Lynch. H. Hagras and V. Callaghan; Embedded Interval type-2 Neuro-Fuzzy Speed Controller for Marine Diesel Engines; Proceedings of the 2006 Information Processing and Management of Uncertainty in Knowledge-based Systems conference, pp. 1340-1347, Paris, France; July 2006

[13] C. Lee, J. Hong, Y. Lin, and W. Lai, "Type-2 Fuzzy Neural Network Systems and Learning," Int. J. of Computational Cognition, vol. 1, pp. 79-90, December 2003

[14] N. Zhang Daryl Beetner, Donald C. Wunsch,Brian Hemmelman and Abul Hasan, “An Embedded Real-Time Neuro-Fuzzy Controller for Mobile Robot Navigation”, IEEE International Conference on Fuzzy Systems, pp. 319-324, 2005

[15] A. Saffioti, “ The uses of fuzzy logic in autonomous robot navigation,”J. Soft Comput., vol. 1, no. 4, pp. 180-197, 1997

[16] T. Fukuda and N. Kubota,” An intelligent robotic system based on fuzy approach, “Proc. IEEE, vol. 87, pp. 1448-1470, Aug. 1999

[17] Nurmaini, S, Zaiton, M, “ An Embedded Fuzzy Type-2 Controller Based Sensor Behavior on Mobile Robot” International Conference on Intelligent System and Application, Vol.2, pp.29-34, Nov, 2008

[18] Nurmaini, S, Zaiton, S, Dayang, N, “Environmental Recognition Using RAM-network based Type-2 Fuzzy Neural for Navigation of Mobile Robot”, International Conference on Computer and Automation Engineering, pp. 296-301 March, 2009

[19] Nesrine Baklouti, A. M. Alimi “Motion Planning in Dynamic and Unknown Environment Using an Interval Type-2 TSK Fuzzy Logic Controller”, IEEE 2007

[20] P. Phokharatul, S. Phaboon, “ Mobile Robot Control Using Type-2 Fuzzy Logic System”, IEEE Conference on Robotics, Atomation and Mechtronics, Singapore, pp 296-299, 2004.

[21] H. Wu and J. Mendel, “Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems,” *IEEE Trans. on Fuzzy Systems*, vol.10, pp. 622-639, October 2002.

[22] D.Wu and W. Tan, “A type-2 fuzzy logic controller for the liquid level process,” in Proc. of the 2004 IEEE Int. Conf. on Fuzzy Systems, Budapest, Hungary, July 2004, vol. 2, pp. 953-958

[23] Q.Liang and J.Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Trans. on Fuzzy Systems*, vol. 8, pp. 535-550, October 2000.