A Low Cost Microcontroller Implementation of Fuzzy Logic Based Hurdle Avoidance Controller for a Mobile Robot

[1]Umar Farooq, [2]K. M. Hasan, [3]Asif Raza, [4]Muhammad Amar, [5]Sabiha Khan, [6]Sameen Javaid
1,3-6Department of Electrical Engineering University of The Punjab Lahore-54590 Pakistan
2Department of Electrical Engineering, University of Engineering & Technology Lahore-54890 Pakistan
[1]engr.umarfarooq@yahoo.com, [2]kmhasan@uet.edu.pk, [3]armalik_87@yahoo.com, [4]amar.ete611@yahoo.com,
[5]sabi412@gmail.com, [6]sameenjavid@hotmail.com

***Abstract*—This paper describes the design of a fuzzy logic based hurdle avoidance controller for mobile robot navigation in noisy and uncertain environments. The employed controller is a two input, two output system. The inputs to the controller are 'distance to hurdle' information from two SRF05 ultrasonic sensors mounted in front of the robot. The outputs from controller are speed commands for two drive wheels. The controller is first designed in MATLAB® and its rule base is optimized for best possible results. It is then implemented using a cheap and readily available AT89C52 microcontroller. The robot with the proposed controller is finally tested in various environments containing obstacles and is found to avoid the obstacles in its way successfully.**

Keywords-Mobile robot navigation;hurdle avoidance;fuzzy logic controller;MATLAB® fuzzy logic toolbox;SRF05 ultrasonic sensors;AT89C52 microcontroller

I. Introduction

Autonomous mobile robots are machines that are capable of moving in an environment containing obstacles. The environment may be structured or unstructured. Since the environment is noisy and unpredictable for most of the times, there is a need for an intelligent controller which is capable of handling these uncertainties. Conventional controller will not work well because of incomplete and imprecise information. Thus a controller with human like reasoning is desired for increasing the probability of success. This leads to the selection of fuzzy logic controller. Fuzzy logic controller (FLC) is a control law that is described by a knowledge based system consisting of IF-THEN rules with vague predicates and a fuzzy logic inference mechanism [1]. Many researchers have utilized the concept of fuzzy modeling in developing behavior based approaches for autonomous navigation of mobile robot [2,3]. In [4], a real time fuzzy logic based navigation controller is implemented which consists of achieving sensor based motion control of a mobile robot among obstacles in structured as well as unstructured environments with collision free motion as the priority. In [5], an experimental neuro-fuzzy controller for sensor-based mobile robot navigation in indoor environments is developed. The autonomous mobile robot uses infrared and contact sensors for detecting targets and avoiding collisions. In [6], a fuzzy control system for navigation of car like robot is modeled by fuzzyTech development environment and is implemented using HC12 Motorola microcontroller. In [7], a fuzzy logic hurdle avoidance controller for navigation of a car like robot is implemented using PIC18F452 microcontroller. Although these approaches give good results in real time, there always remains a need for lowering the system cost further.

In this paper, fuzzy logic based hurdle avoidance controller for differentially steered mobile robot is designed using MATLAB® and is implemented with a low cost AT89C52 microcontroller. Experimental results have shown the validity of fuzzy reasoning in generating collision free path for robot motion amongst the obstacles.

II. Brief Overview of Method

In robotics design, it is often desirable to have a robot act like a human being than a machine. Traditional logic is based on two values, one (true) and zero (false). This is inadequate for approximating the human decision making process. Fuzzy logic uses the entire interval between zero and one and can therefore be used to closely mimic human reasoning [8,9]. The design process of a fuzzy logic system can be divided into three stages:

- Fuzzification,
- Fuzzy Inference Engine, and
- Defuzzification

In fuzzification stage, a quantified numerical control variable is converted into a qualitative value like small, medium or large known as linguistic variables. These linguistic variables are described by a membership function which has a value between zero and one. The computationally less intensive membership functions are triangular and trapezoidal. Once the inputs are fuzzified, they are fed to fuzzy inference engine. This engine consists of two sub-blocks namely fuzzy rule base and fuzzy implication. The fuzzy rule base is a set of if-then statements known as linguistic rules which describes the behavior of the robot for a particular set of inputs. Fuzzy implication evaluates the set of rules to compute a qualitative output result for the controller. The commonly used implication method is Mamdani implication. This stage plays a key role on the fuzzy controller operation as its rules model the whole behavior of the robot. Finally, the qualitative value is converted into numerical value in defuzzification stage. There are many defuzzification methods available. Among them, Center of Area (COA) method is easy to implement and has a fast execution time.

III. Fuzzy Controller Design

A two input, two output fuzzy logic controller is designed for hurdle avoidance task. The distance information

978-1-4244-5539-3/10/$26.00 ©2010 IEEE

from left and right ultrasonic sensors forms the input to fuzzy controller while two outputs from the controller are pulse width modulated signals to control left and right motor speeds. MATLAB® Fuzzy Logic Toolbox is used to aid in FLC design. The toolbox contains functions, graphical user interfaces and data structures that allow the user to quickly design, test, simulate and modify a fuzzy inference system [10]. The block diagram of system is shown in Fig. 1. The steps involved in fuzzy controller design are described in this section.

A. Fuzzification

The input to FLC i.e., the 'distance form hurdle' information from left sensor (LS) and right sensor (RS) is described by three linguistic variables: Near, Med and Far; the universe of discourse being from zero (minimum) to hundred (maximum) for each sensor. The output of FLC i.e., left motor speed (LM) and right motor speed (RM) is also described by three linguistic values: Slow, Med and Fast; the universe of discourse being from zero (minimum) to hundred (maximum) which describes the duty cycle of PWM signal. All the linguistic variables are described by triangular membership function, shown in Fig. 2. Triangular membership function (MF) is selected because of limited computational resources of microcontroller. In general, it is specified by three parameters {a,b,c}[11]:

$$triangle(x;a,b,c)=\begin{cases}0, x<a\\ \dfrac{x-a}{b-a}, a\le x\le b\\ \dfrac{c-x}{c-b}, b\le x\le c\\ 0, c\le x\end{cases} \qquad (1)$$

By using min and max, an alternate expression can be described as:

$$triangle(x;a,b,c)=\max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) \qquad (2)$$

The parameters {a,b,c} (with a<b<c) determine the x coordinates of three corners of the underlying membership function.

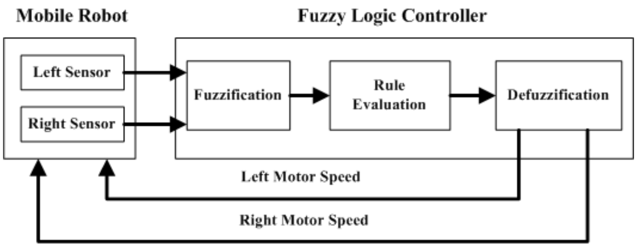
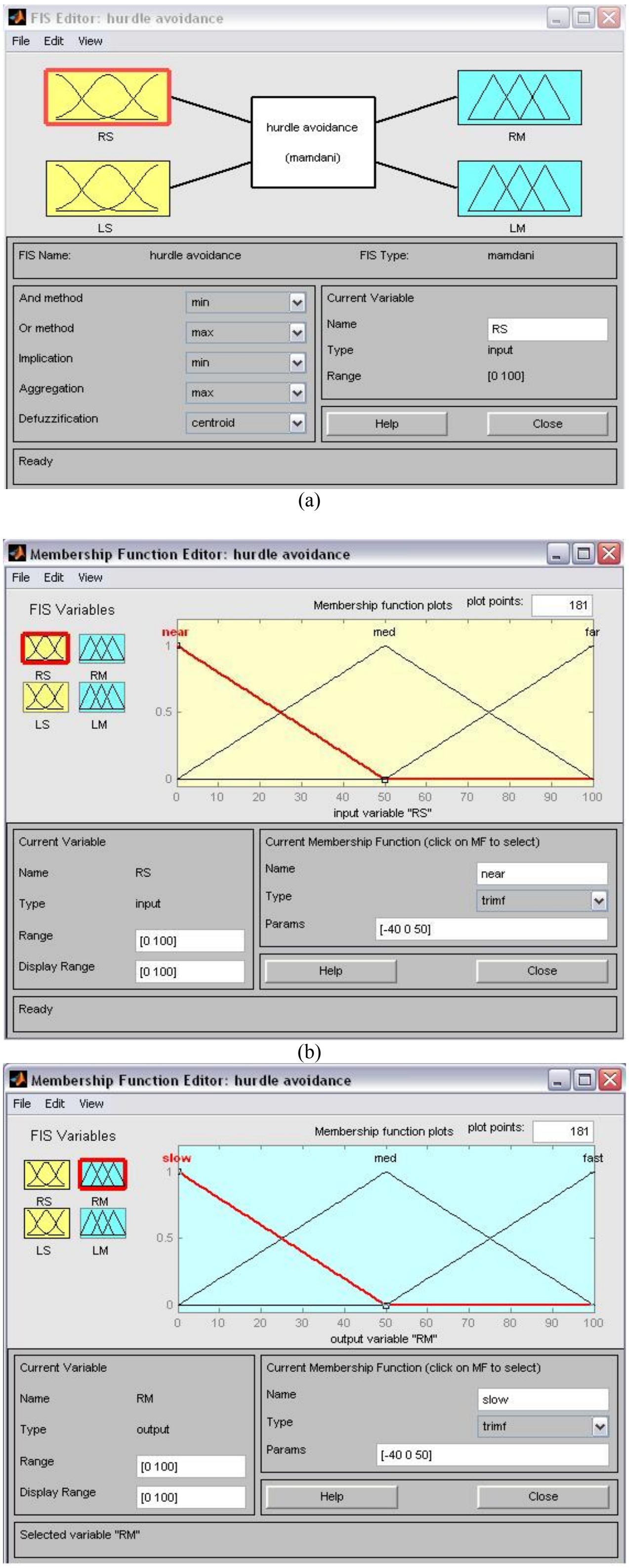


Figure 1. System Block Diagram



FIS Name: hurdle avoidance
FIS Type: mamdani
And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid
Current Variable
Name: RS
Type: input
Range: [0 100]

(a)

plot points: 181
Current Variable
Name: RS
Type: input
Range: [0 100]
Display Range: [0 100]
Current Membership Function (click on MF to select)
Name: near
Type: trimf
Params: [-40 0 50]

(b)

plot points: 181
Current Variable
Name: RM
Type: output
Range: [0 100]
Display Range: [0 100]
Current Membership Function (click on MF to select)
Name: slow
Type: trimf
Params: [-40 0 50]

Selected variable "RM"

(c)

Figure 2. (a) FLC GUI (b) Input MF (c) Output MF

B. *Fuzzy Rule Base*

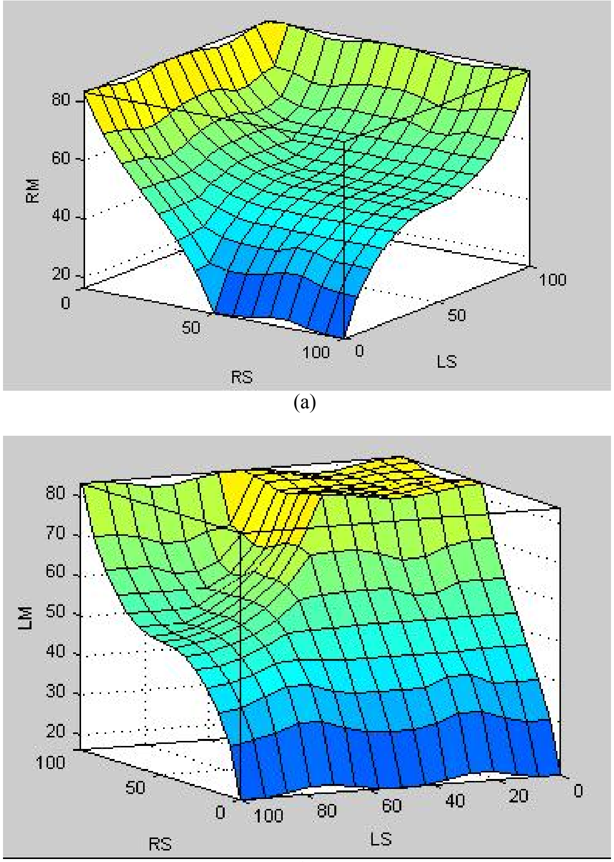
Nine rules are designed for controlling robot motion in an environment containing obstacles. Of these rules, only two are fired at any instant. These rules establish the relation between sensor values and motor speeds in terms of linguistic values. A control surface is then plotted to visualize the variation in motor speeds given the sensor values. The rule base for right and left motor speed control is shown in Table 1 and Table 2 respectively while control surface is shown in Fig. 3.

TABLE I. RULE BASE FOR CHANGE IN RIGHT MOTOR SPEED

RS/LS	Near	Med	Far
Near	Fast	Fast	Fast
Med	Slow	Med	Fast
Far	Slow	Med	Fast

TABLE II. RULE BASE FOR CHANGE IN LEFT MOTOR SPEED

RS/LS	Near	Med	Far
Near	Slow	Slow	Slow
Med	Fast	Fast	Med
Far	Fast	Fast	Fast



(a)

(b)

Figure 3. (a) Control Surface for Right Motor (b) Control Surface for Left Motor

C. *Fuzzy Implication*

Fuzzy implication helps to evaluate the consequent part of each rule. Among the various implication methods available in literature, Mamdani implication method is selected. After the inputs have been fuzzified and FLC know the degree to which each part of the antecedent of a rule has been satisfied, degree of fulfillment of (DOF) of each rule is calculated using AND operator. The output membership function is then truncated at DOF level. All the rules are evaluated in this manner and final output membership functions are aggregated in a cumulative manner using OR operator to yield the final fuzzy output. An example of the implication process is shown in Fig. 4 where right sensor measures 'far' while left sensor measures 'near'.

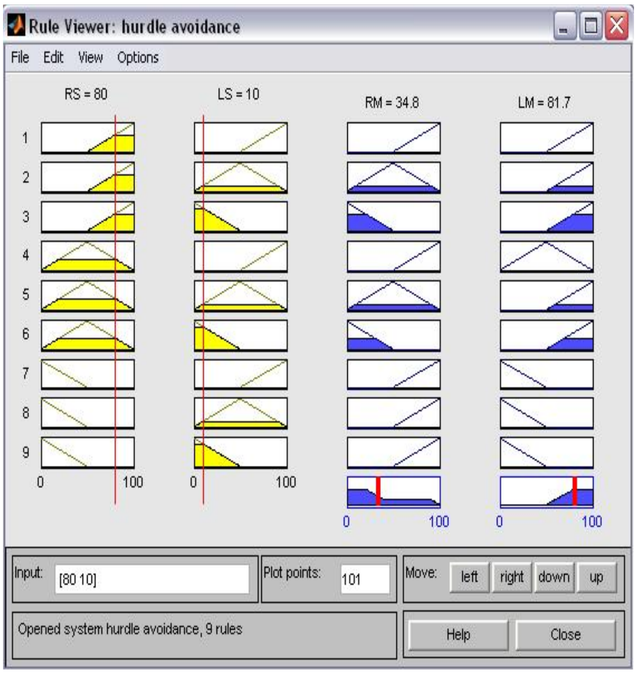
D. *Defuzzification*

The result of the implication and aggregation step is the fuzzy output, which is the union of all the individual rules that are validated or fired. Conversion of this fuzzy output to crisp output is defined as defuzzification. COA method has been used for this purpose and is described as:

$$Z_o = \frac{\sum_{i=1}^{n} Z_i \mu_{out}(Z_i)}{\sum_{i=1}^{n} \mu_{out}(Z_i)}$$

(3)

where $\mu_{out}(Z_i)$ are the i=1,2,…,n sampled values of the aggregated output membership function and Z_o is the crisp value which describes the duty cycle of PWM signal for controlling the speed of motors.



RS = 80

LS = 10

RM = 34.8

LM = 81.7

Input: [80 10]

Plot points: 101

Opened system hurdle avoidance, 9 rules

Figure 4. Implication and Defuzzification Process

IV. Fuzzy Controller Implementation

FLC is implemented in three different ways to improve the response time and resulting efficiency. These configurations include single microcontroller implementation, multiple microcontrollers' implementation and microcontroller with external EEPROM.

A. *Single Chip Microcontroller Implementation*

In single chip microcontroller implementation, the algorithm first computes the PWM duty cycle value for right motor after fuzzification, rule base evaluation and defuzzification. The same steps are then repeated to calculate the PWM duty cycle value for left motor. The block diagram for single chip microcontroller implementation is shown in Fig. 5. The algorithm is implemented in C language using Keil development platform. The inputs to algorithm are defined in range [0,100] and outputs in range [0,50]. The main steps are described below:

1) Set the PWM frequency equal to 100Hz and initialize the PWM duty cycle value to 100%. Also, initialize an array of 50 characters with zeros to store final fuzzy output value.
2) Trigger the left and right sensors for reading the distance information. Scale the distance information such that it lies between zero and hundred.
3) Calculate the degree of membership of the defined linguistic variables (near, med, far) for each sensor value. The degree of membership is defined in range from zero to hundred instead of zero to one to prevent floating point arithmetic thereby reducing the computational burden on microcontroller. The calculation is done using 'unsigned char' data-type.
4) Fire the rules in sequence. First, calculate the DOF of 1st rule which is taken as the minimum value amongst the degree of memberships of two linguistic values in antecedent part of that rule. The output linguistic variable for that rule which is described by equation of line (membership function) is then cut at DOF level and a loop is started from index defined for that membership function to store the result of firing that rule. At each index, equation of line for output membership function is evaluated. If computed value falls below DOF level and above already stored value, then that value is stored in output array at that index. If computed value is greater than or equal to DOF value and DOF value is greater than already stored value at that index, then DOF value is stored at that index of array. These two comparisons are performed until the final index value for particular output membership function is reached. This complete step is repeated for all the rules in rule base. The value which is stored after the execution of step (4) is final fuzzy value for right motor speed.
5) Perform defuzzification of output fuzzy value using COA method. A loop is started to sum the product of every array index and value at that index. The resultant value is then divided by sum of values at all the indexes of array to calculate crisp output. This calculation is performed using 'unsigned integer' data-type. The output crisp value is multiplied by 2 to generate the PWM duty cycle value for right motor. This scaling is done because the outputs are defined in range [0,50] to overcome the data memory limitations.
6) Repeat steps 2 through 5 to compute PWM duty cycle value for left motor.

B. *Mutiple Chips Microcontroller Implementation*

In multiple chips microcontroller implementation, two microcontrollers are employed for generating PWM duty cycle values for left and right motors in parallel. The same algorithm as described for single chip implementation is performed in each microcontroller. Experimental results have shown that this technique has better response time. The block diagram of this implementation is shown in Fig. 6. Another implementation involves adding third microcontroller which is dedicated for generating speed commands for both motors after taking crisp value from each microcontroller. Advantage which lies in using multiple chip microcontroller implementation is modularity i.e., other modules can be added easily without changing the system much thereby reducing the overall complexity.

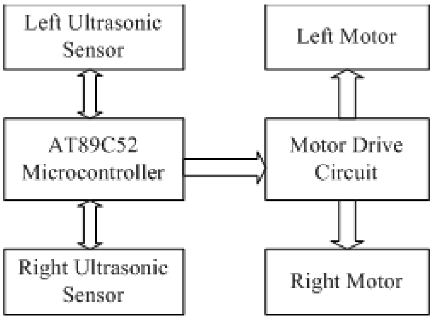


Figure 5. Single Chip Microcontroller Implementation

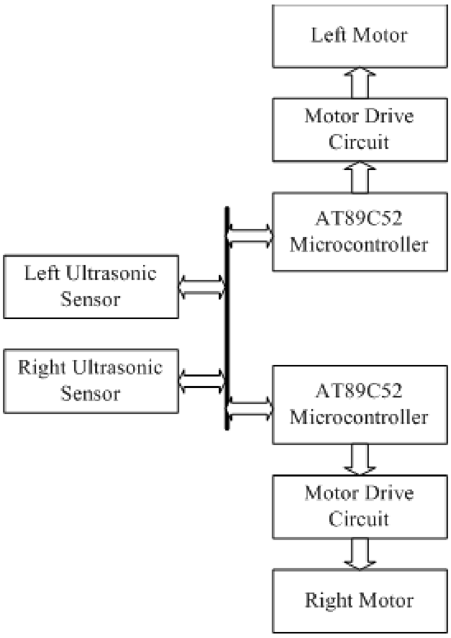
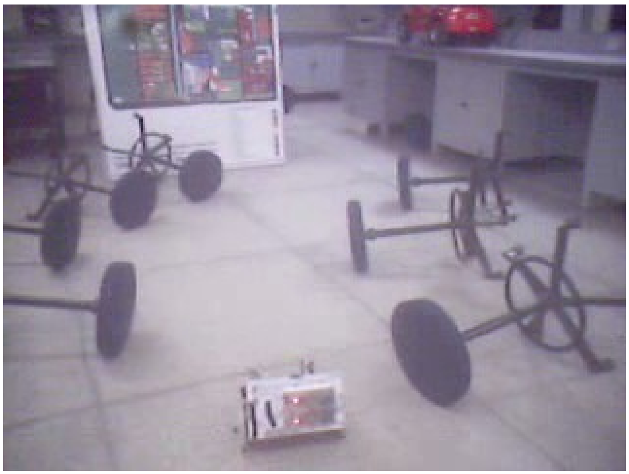


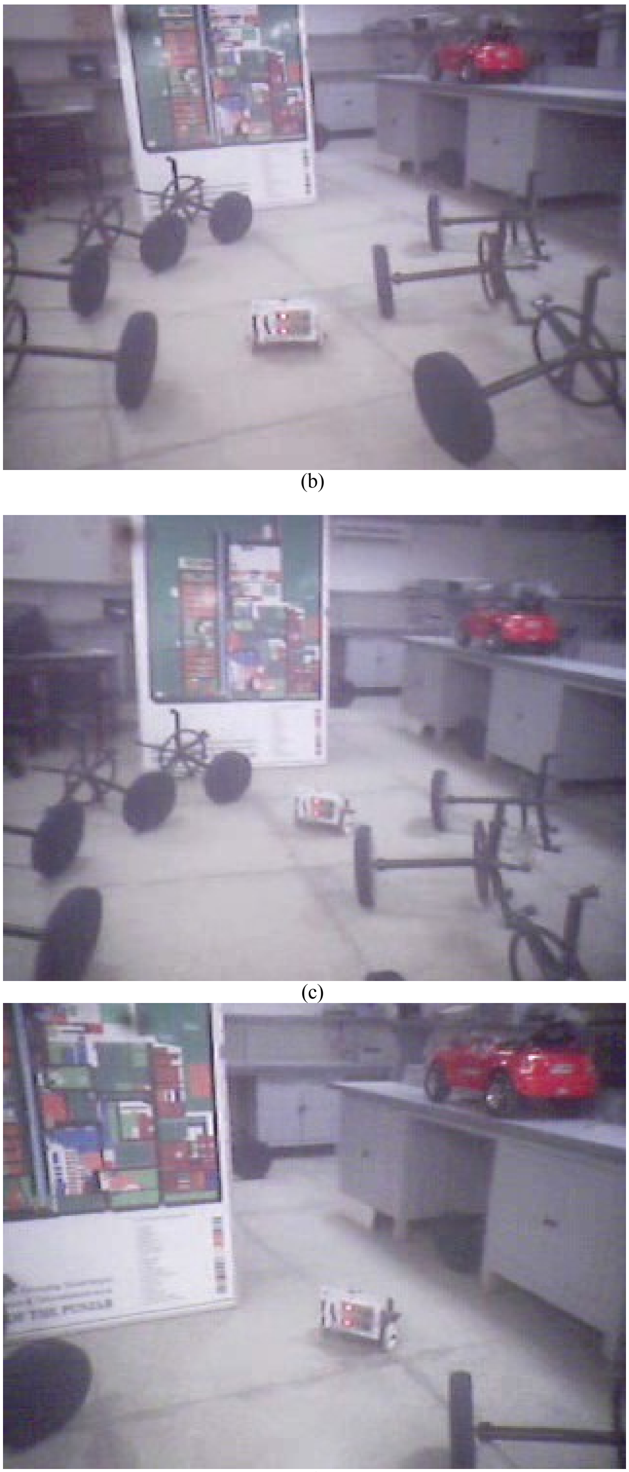
Figure 6. Multiple Chips Microcontroller Implementation

V. EXPERIMENTAL RESULTS

Results are performed using a differentially steered mobile robot in an environment containing obstacles. The resulting motion of robot among obstacles is shown in Fig. 7.



(a)



(b)

(c)

(d)

Figure 7. (a) - (d) Hurdle Avoidance Steps

VI. CONCLUSIONS

In this paper, design of a fuzzy controller for hurdle avoidance in complex environments is presented. The designed controller is tested in real time using a low cost

AT89C52 microcontroller. Experimental results obtained after testing the controller has proved the validity of system.

REFERENCES

[1] Ronald R. Yager and Dimitar P. Filev, Essentials of Fuzzy Modeling and Control, Johan Wiley & Sons, 2002.

[2] Anis Fatmi, Amur Al Yahmadi, Lazhar Khiji, and Nouri Masmoudi, "A fuzzy logic based naviation of a mobile robot," International Journal of Applied Mathematics and Computer Sciences, vol. 1, no. 2, pp. 87-92, August 2006.

[3] M. K. Singh, D. R. Parhi, S. Bhowmik, and S. K. Kashyap, "Intelligent controller for mobile robot: fuzzy logic approach," Proc. 12th International Conference of International Association for Computer Method and Advances in Geomechanics, Goa, India, October 2008.

[4] Thamar E. Mora, and Edgar N. Sanchez, "Fuzzy logic based real time navigation controller for a mobile robot," Proc. International Conference on Intelligent Robots and Systems, pp. 612-617, October 1998.

[5] Petru Rusu, Emil M. Petriu, Thom E. Whalen, Aurel Cornell, and Hans J. W. Spoelder, " Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation," IEEE Transactions On Instrumentation and Measurement, vol. 52, no. 4, August 2003.

[6] Yordan Chervenkov and Ovid Farhi, "Fuzzy control system with implementation on an embedded microcontroller," Proc. 2nd ISCCSP, March 13-15, 2006.

[7] Robert T. Casey and Mike Hensler, "Fuzzy steering for autonomous MCU-based mobile robots," Proceedings of the 5th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, pp. 211-216, 2006.

[8] A. C. Nearchou, "Adaptive navigation of autonomous vehicles using evolutionary algorithms," Artificial Intelligence in Engineering, vol. 13, pp. 159-173, 1999.

[9] S. K. Harisha, Ramkanth Kumar P., M. Krishma, and S. C. Sharma, "Fuzzy logic reasoning to control mobile robot on pre-defined strip path," Proc. World Academy of Science, Engineering and Technology, vol. 32, pp. 729-733, August 2008.

[10] MATLAB® Fuzzy Logic Toolbox

[11] Vasmi Mohan Peri, Fuzzy logic controller for an autonomous mobile robot, M. S. Thesis, Cleveland State University, May 2002.