

FUZZY LOGIC CONTROL FOR AN AUTONOMOUS ROBOT

Vamsi Mohan Peri

*Department of Electrical and Computer Engineering
Cleveland State University
Cleveland, Ohio, USA*

Vamsi291@rediffmail.com

Dan Simon

*Department of Electrical and Computer Engineering
Cleveland State University
Cleveland, Ohio, USA*

d.j.simon@csuohio.edu

Abstract – In this paper an autonomous wall-following robot is presented. The inputs are obtained from ultrasonic sensors mounted. These inputs are sent to a Microchip PIC16F877 microcontroller onboard the robot, which analyses the data and provides the necessary control signal. A fuzzy logic controller is used to control the robot's motion along the predefined path. The robot was first modeled in Matlab Simulink and the fuzzy logic rules were optimized for the best results possible. Later the microcontroller was programmed in C language using a PCW C-compiler and tested. Experimental results are presented to show the performance of the controller.

I. INTRODUCTION

Mobile robots are mechanical devices capable of moving in an environment with a certain degree of autonomy. Autonomous navigation is associated with the availability of external sensors that capture information from the environment through visual images, or through distance or proximity measurements. The most common sensors are distance sensors (ultrasonic, laser, etc.) capable of detecting obstacles and of measuring the distance to walls close to the robot path. When advanced autonomous robots navigate within indoor environments (industrial or civil buildings), they have to be endowed with the ability to move through corridors, to follow walls, to turn corners and to enter open areas of the rooms.

In attempts to formulate approaches that can handle real world uncertainty, researchers are frequently faced with the necessity of considering tradeoffs between developing complex cognitive systems that are difficult to control, or adopting a host of assumptions that lead to simplified models which are not sufficiently representative of the system or the real world. The latter option is a popular one which often enables the formulation of viable control laws. However, these control laws are typically valid only for systems that comply with imposed assumptions, and that operate in small neighbourhoods of some nominal state. The option that involves complex systems has been less prevalent due to the lack of analytical methods that can adequately handle uncertainty and concisely represent knowledge in practical control systems. Recent research and application employing non-analytical methods of computing such as fuzzy logic, evolutionary computation, and neural

networks have demonstrated the utility and potential of these paradigms for intelligent control of complex systems. In particular, fuzzy logic has proven to be a convenient tool for handling real world uncertainty and knowledge representation.

A. Literature Survey

As regards the corridor and wall-following navigation problem, some control algorithms based on artificial vision have been proposed. In [1], image processing is used to detect perspective lines to guide the robot along the center axis of the corridor. In [2], two lateral cameras mounted on the robot are used, and the optical flow is computed to compare the apparent image velocity on both cameras in order to control robot motion. In [3, 4], one camera is used to drive the robot along the corridor axis or to follow a wall, by using optic flow computation and its temporal derivatives. In [5], a globally stable control algorithm for wall-following based on incremental encoders and one sonar sensor is developed. In [6], a theoretical model of a fuzzy based reactive controller for a non-holonomic mobile robot is developed. In [7], an ultrasonic sensor is used to steer an autonomous robot along a concrete path using its edged as a continuous landmark. In [8], a mobile robot control law for corridor navigation and wall following based on sensor and odometric sensorial information is proposed.

B. Paper Layout

In Section II the mathematical model of the robot is developed. In Section III the fuzzy logic controller is developed and discussed in detail. In Section IV the hardware and software design of the robot are discussed in detail. Performance comparisons are given in Section V to demonstrate the efficiency of the proposed method. Finally the conclusions drawn from the results obtained are given in Section VI.

II. MATHEMATICAL MODEL

A. Objective

The robot that is to be controlled was initially built to take part in an IEEE competition, held at Cleveland State University in 2004. The goal of this competition is to compete head to head on a playing board with an opponent

and obtain the most points in an allotted amount of time. The dimensions of the robot had to be within those specified in the competition. The robot had to fit within a 1.5-foot square and could not exceed 1.5 feet in height. It had to be totally autonomous, couldn't transmit or receive signals to or from the outside of the playing area, and could not be equipped to intentionally harm its opponents. It also couldn't carry any onboard cameras. The main goal of this work, keeping in mind the requirements of the competition, was to cover as much of the playing area as possible within the shortest time so that the maximum points could be scored. The robot that was built placed second among eleven other competitors from different schools.

This type of robot has many applications, including habitat mapping, environmental monitoring, mapping and surveillance, search and rescue operations, planetary surface exploration, agriculture, manufacturing, and service.

B. Types of Robotic Behaviors and Controllers

A mobile robot could be modeled in numerous ways, but the most important factor for defining the model would be the application and the complexity involved. The mobile robot designed in this work is a wheeled robot intended for indoor use as opposed to other types (legged, airborne, and submersible mobile robots). This robot type is the easiest to model, control, and build. There are various behaviours that could be modeled, like wall following, collision avoidance, corridor following, goal seeking, adaptive goal seeking, etc. With the competition in mind we thought of implementing a wall following robot. This robot would follow the boundaries of the playing area and cover a maximum area in a predefined path programmed into its onboard microcontroller.

Various control techniques have been proposed and are being researched. The control strategies of mobile robots can be divided into open loop and closed loop feedback strategies. In open loop control, the inputs to the mobile robots (velocities or torques) are calculated beforehand, from the knowledge of the initial and end position (and of the desired path between them in the case of path following). This strategy cannot compensate for disturbances and model errors. Closed loop strategies, however, may give the required compensation since the inputs are functions of the actual state of the system and not only of the initial and end points. Therefore disturbances and errors causing deviations from the predicted state are compensated by the use of the inputs. Of the many available closed loop control systems, including P (proportional) control, PI (proportional integral) control, and PID (proportional integral derivative) control, fuzzy logic control was selected as it was easiest to implement for a highly nonlinear robot model.

Although a relatively new concept, fuzzy logic is being used in many engineering applications because it is considered by designers to be the simplest solution available for the specific problem. What gives fuzzy logic advantages

over more traditional solutions is that it allows computers to reason more like humans, responding effectively to complex inputs to deal with linguistic notions such as 'too hot', 'too cold' or 'just right'. Furthermore, fuzzy logic is well suited to low-cost implementations based on cheap sensors, low-resolution analog-to-digital converters, and 4-bit or 8-bit microcontroller chips. Such systems can be easily upgraded by adding new rules to improve performance or add new features. In many cases, fuzzy control can be used to improve existing traditional control systems by adding an extra layer of intelligence to the current control method. In many cases, the mathematical model of the system to be controlled may not exist, or may be too "expensive" in terms of computer processing power and memory, and a system based on empirical rules may be more effective. Implementing the enhanced, traditional P controller can be a challenge, especially if auto-tuning capabilities to help find the optimal P constants are desired. However, the theory of P control is very well known and widely used in many other control applications. On the other hand, fuzzy control seems to accomplish better control quality with less complexity (if tuning or gain scheduling is needed for the P approach). Approximation of the second-order switching curve used in time-optimal control systems by a polynomial of the first or higher order makes fuzzy control a better candidate for time-optimal control applications [10]. As a relatively new control method, fuzzy logic provides more space for further improvements.

Originally advocated by Zadeh [11] and Mamdani and Assilian [12], fuzzy logic has become a means of collecting human knowledge and experience and dealing with uncertainties in the control process. Now fuzzy logic is becoming a very popular topic in control engineering. Considerable research and applications of this new area for control systems have taken place. Control is by far the most useful application of fuzzy logic theory, but its successful applications to a variety of consumer products and industrial systems have helped to attract growing attention and interest.

Based on its design simplicity, its ease of implementation, and its robustness properties, a fuzzy logic controller is used in this paper to control the navigation behaviour of an autonomous mobile robot.

C. Equations defining the Robot

For a differential drive the kinematics equations in the world frame are as follows [13].

$v_r(t)$ = linear velocity of right wheel

$v_l(t)$ = linear velocity of left wheel

$w_r(t)$ = angular velocity of right wheel

$w_l(t)$ = angular velocity of left wheel

r = nominal radius of each wheel

L = distance between the two wheels

R = instantaneous curvature radius of the robot trajectory, relative to the mid-point of the wheel axis

ICC = Instantaneous Center of Curvature
 $R - (L/2)$ = Curvature radius of trajectory described by left wheel
 $R + (L/2)$ = Curvature radius of trajectory described by right wheel

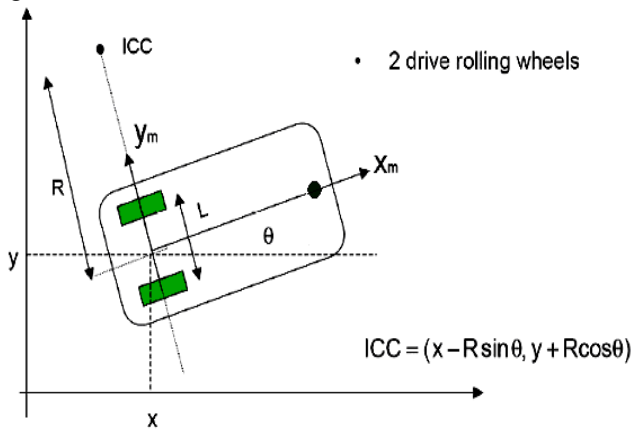


Fig. 1 Kinematic model of the robot

Fig. 1 shows the variables used in the kinematic and dynamic equations discussed below. With respect to ICC the angular velocity of the robot is given as follows

$$w(t) = v_r(t)/(R + (L/2)) \quad (1)$$

$$w(t) = v_r(t)/(R - (L/2)) \quad (2)$$

$$w(t) = (v_r(t) - v_l(t))/L \quad (3)$$

The instantaneous curvature radius of the robot trajectory relative to the mid-point of the wheel axis is given as

$$R = (L(v_l(t) + v_r(t)))/(2(v_l(t) - v_r(t))) \quad (4)$$

Therefore the linear velocity of the robot is given as

$$v(t) = w(t)R = (v_r(t) + v_l(t))/2 \quad (5)$$

The kinematics equations in the world frame can be represented as follows.

$$\dot{x}(t) = v(t) \cos \theta(t) \quad (6)$$

$$\dot{y}(t) = v(t) \sin \theta(t) \quad (6)$$

$$\dot{\theta}(t) = w(t)$$

This implies

$$x(t) = \int_0^t v(\sigma) \cos(\theta(\sigma)) d\sigma$$

$$y(t) = \int_0^t v(\sigma) \sin(\theta(\sigma)) d\sigma \quad (7)$$

$$\theta(t) = \int_0^t w(\sigma) d\sigma$$

The above equations can also be represented in the following form.

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} v(t) \cos \theta \\ v(t) \sin \theta \\ w(t) \end{bmatrix} \begin{bmatrix} v_l(t) \\ v_r(t) \end{bmatrix}$$

$$= \begin{bmatrix} ((v_r + v_l) \cos \theta)/2 \\ ((v_r + v_l) \sin \theta)/2 \\ (v_r - v_l)/L \end{bmatrix}$$

$$= \begin{bmatrix} \cos \theta/2 & \cos \theta/2 \\ \sin \theta/2 & \sin \theta/2 \\ -1/L & 1/L \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix} \quad (8)$$

These are the equations that are used to build a model of the robot. These equations were used to simulate the robot in MATLAB Simulink. The fuzzy logic controller was tested and fine tuned on this model for optimum results, as well as compared with other controllers.

III. FUZZY LOGIC CONTROLLER

The FLC used has two inputs: error in the position and error in the angle of the robot. Thus the FLC is a two input, two output system.

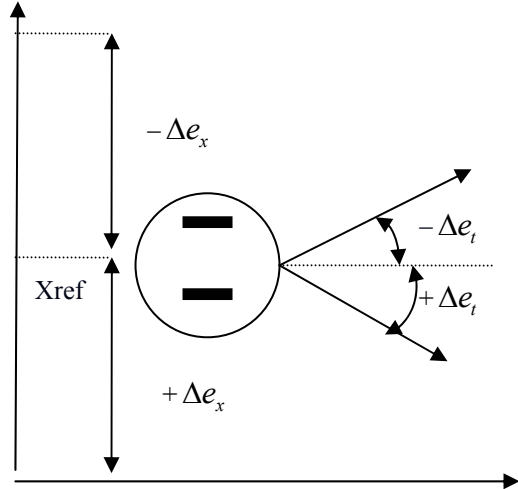


Fig. 2 Robot in Cartesian space.

Fig. 2 shows the robot in Cartesian space. It is clear that the two inputs are error in angle of orientation θ and error in distance x . The output of the controller (that is, the control signals) would be pulse-width-modulated signals to control the angular velocity of the two servo wheels. The block diagram of the robotic system is given in Fig. 3.

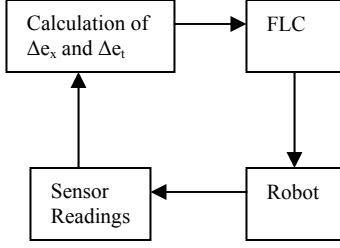


Fig. 3 Block diagram of the robotic system.

A. Fuzzy Inference System

In the fuzzy inference system designed in our system, the membership functions of the linguistic variables are sum normal triangular functions. That means that the sum of the membership functions of any variables at any given point in the domain is equal to one. The triangular membership functions are used for their simplicity as is quite commonly done. Calculating fuzzified inputs and the areas of these functions is simple and fast when compared to other membership functions such as Gaussian. The rule base for the FLC proposed in this study is listed in Tables I and II below. Thus we see that there are 18 total rules for the two wheels combined. Of these rules, at any instant only two given rules are fired thus making it easier to understand and debug the rules.

TABLE I
RULE BASE FOR Δw_r .

$\Delta e_x / \Delta e_t$	N(negative)	Z(zero)	P(positive)
N(negative)	S(slow)	S(slow)	S(slow)
Z(zero)	S(slow)	F(fast)	M(medium)
P(positive)	S(slow)	M(medium)	F(fast)

TABLE II
RULE BASE FOR Δw_l .

$\Delta e_x / \Delta e_t$	N(negative)	Z(zero)	P(positive)
N(negative)	F(fast)	M(medium)	S(slow)
Z(zero)	M(medium)	F(fast)	S(slow)
P(positive)	S(slow)	S(slow)	S(slow)

B. Sum Normal fuzzification [14]

Consider the i th fuzzy membership function of the j th input z_j . Its modal point is denoted as c_{ij} , its lower half width as b_{ij}^- , and its upper half width as b_{ij}^+ . The membership function attains a value of 1 when the input is c_{ij} . As the input decreases from c_{ij} , the membership function value decreases linearly to 0 at $c_{ij} - b_{ij}^-$ and remains at 0 for all inputs less than $c_{ij} - b_{ij}^-$. As the input increases from c_{ij} , the membership function value decreases linearly to 0 at $c_{ij} + b_{ij}^+$, and remains at 0 for all inputs greater than $c_{ij} + b_{ij}^+$. The membership function of the j th crisp input z_j in its i th fuzzy set is given by

$$f_{ij}(z_j) = \begin{cases} 1 + (z_j - c_{ij}) / b_{ij}^- \\ 1 - (z_j - c_{ij}) / b_{ij}^+ \\ 0 \end{cases} \quad (9)$$

Our fuzzy system has two outputs of which we are considering only one output in this section for notational convenience. Suppose there are a total of M rules in the FLS. The consequent of the j th rule is a triangular fuzzy set with modal point γ_j , lower half-width β_j^- , and upper half width β_j^+ . That is, the fuzzy set of the consequent of the j th rule is given as

$$m_j(y) = \begin{cases} 1 + (y - \gamma_j) / \beta_j^- \\ 1 - (y - \gamma_j) / \beta_j^+ \\ 0 \end{cases} \quad (10)$$

Suppose that the j th rule is a consequent of z_1 belonging to fuzzy set i and z_2 belonging to fuzzy set k . Then the activation level of the consequent of the j th rule is w_j which is given as

$$w_j = \min[f_{i1}(z_1), f_{k2}(z_2)] \quad (11)$$

So the fuzzy output when z_1 belongs to fuzzy set i and z_2 belongs to fuzzy set k is given as

$$\bar{m}_j(y) = w_j m_j(y) \quad (12)$$

The overall fuzzy output $m(y)$ takes into account the possibility that each input falls into more than one fuzzy set so more than one rule can be fired at the same time.

$$m(y) = \sum_{j=1}^M \bar{m}_j(y) \quad (13)$$

The fuzzy output is mapped to a crisp number y using centroid defuzzification.

$$\hat{y} = (\sum_{j=1}^M w_j \Gamma_j J_j) / (\sum_{j=1}^M w_j J_j) \quad (14)$$

Γ_j and J_j are the centroid and the area of the j th output fuzzy membership function. The centroid of $m_j(y)$, the j th output fuzzy set, is defined as

$$\Gamma_j = \frac{\int y m_j(y) dy}{\int m_j(y) dy} \quad (15)$$

After substituting (10) into the above equation we obtain

$$\Gamma_j = \frac{\beta_j^+ (3\gamma_j + \beta_j^+) + \beta_j^- (3\gamma_j - \beta_j^-)}{3(\beta_j^+ + \beta_j^-)} \quad (16)$$

Thus the final fuzzy output is \hat{y} which is the change in the speed of the right wheel in rad/sec. Similarly the change in the speed of the left wheel is also calculated.

IV. DESIGN OF THE ROBOT

The robot built for the IEEE competition is shown in Fig. 4 below. Its dimensions are 14 inches in length and 12 inches in width.

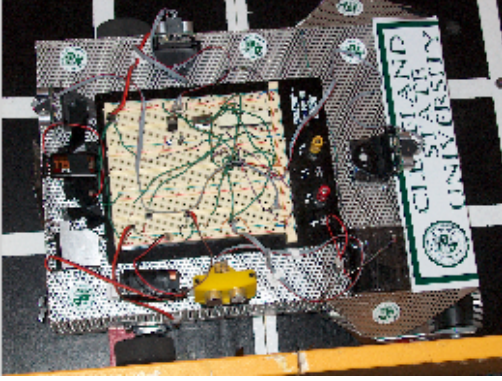


Fig. 4 Picture of the robot.

A. Hardware Model Description

The robot has two wheels on its sides which are driven separately by two modified servo motors, and there is a roller wheel in front for balance as well as for smooth turning. The two servo motors used are MX-50Hp manufactured by MPI and are rated at 4.8 VDC. These are run using PWM signals generated from the PIC16F877 microcontroller. A pulse with a 15 ms time period and an on-time of 1500 us will stop the rotation of the motors. If the on-time of the pulse is changed from 1500 to 1000 us the wheel rotates with increasing velocity in the clockwise direction. Similarly if the on-time of the pulse is changed from 1500 to 2000 us the wheel rotates with increasing velocity in the anticlockwise direction.

The sensors used on the robot are SRF04 ultrasonic range finder, two on a side and one in the front. The transmitter works by transmitting a pulse of sound outside the range of human hearing at 40 KHz. This pulse travels at the speed of sound away from the ranger in a cone shape and the sound reflects back to the ranger from any object in the path of sonic wave. If received back at the ranger, the ranger reports this echo to the microcontroller and the microcontroller can then compute the distance to the object based on the elapsed time.

Due to the hardware limitations it is not possible to achieve good robot control results by doing the fuzzy calculations on board the autonomous robot, as the system clock is only 4 MHz. This results in about 0.4 seconds to complete the whole process of fuzzification and defuzzification, and thus the response of the system in real time is too slow to allow the robot to attain a steady state. Thus we have used a lookup table instead for fuzzy logic control, and also two PIC16F77 chips are used. One is exclusively used to generate the PWM signals to run the two servo motors. The other microcontroller is connected to the sensors and calculates the present errors in distance and angle and sends the necessary control signals to the microcontroller connected to the servo motors. Figure 5 shows a block diagram of the hardware setup.

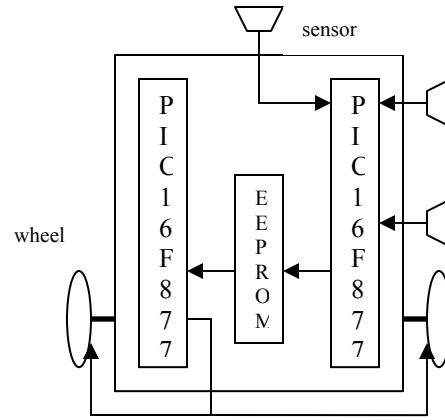


Fig. 5 Block diagram of hardware used.

B. Simulink Model Description

The Simulink model is shown in Fig. 6. The fuzzy block in the Simulink model is a customized Matlab m-file function block replacing Matlab's fuzzy logic toolbox. The output variables X , Y , and T are the two Cartesian coordinates of the robot position, along with its angular orientation. The X_{ref} input is intended to keep the robot a certain distance away from the wall which it is following. The T_{ref} input is zero, as it is intended that the orientation of the robot be maintained at a zero angle relative to the wall. The process and measurement noise are shown in the Simulink block diagram as white noise.

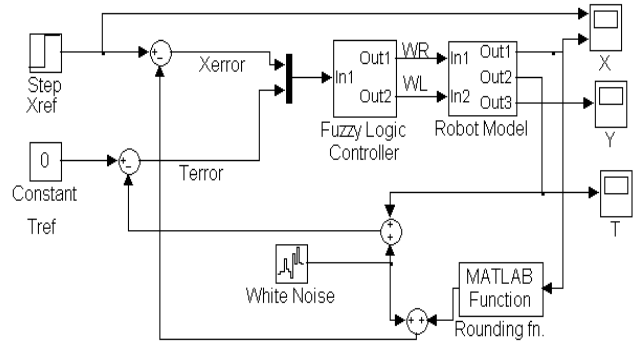


Fig. 6 Simulink model of the robot.

V. EXPERIMENTAL RESULTS

The plots in Figs. 7 and 8 shown below compare the simulation results obtained of the response of the robot both with a FLC and a P controller. The P controller was tuned for optimum results using a genetic algorithm. The robot is asked to maintain a distance of 4 inches from the wall. It is seen that the robot starts from zero and aligns itself to the reference distance. Fig. 7 shows the angle of orientation of the robot as a function of time. Fig. 8 shows the distance from the wall as the robot tries to reach the reference distance. The solid lines represent the response

of the robot with a fuzzy logic controller and the dashed lines represent the response of the robot with a P controller. It is obvious that the FLC is performing much better than the P controller, as it is seen that the robot reaches a steady state much earlier with a FLC than a P controller.

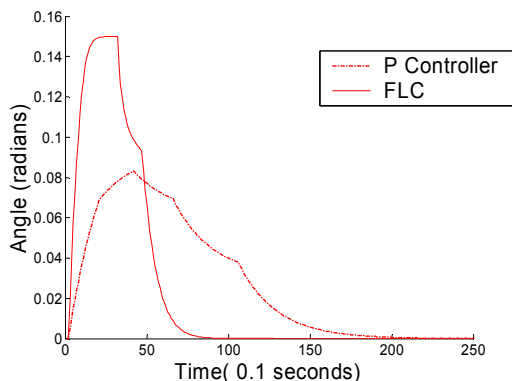


Fig. 7 Angle of orientation as a function of time (time in steps of 0.1s)

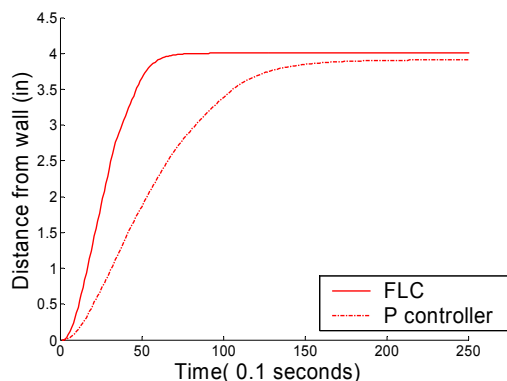


Fig. 8 Distance from the wall as a function of time (t in steps of 0.1s).

The results above are those obtained with computer simulations in Simulink. These were verified to be comparable with those obtained during real time implementation / experimentation with the PIC16F877 microcontroller. It was also observed that due to speed limitations of the microcontroller, the response was comparable only when a lookup table was used to implement the FLC, rather than calculating the control signals using the FLC equations.

VI. CONCLUSIONS

In this paper a fuzzy logic controller to control the motion of differential drive mobile robots has been presented. Simulations were carried out on a non-holonomic mobile robot to test the performance of the proposed fuzzy controller. The mobile robot was seen to reach any end

position in an efficient manner, and whatever its initial configuration in the x - y Cartesian plane. Thus a Fuzzy Logic Controller has been implemented in real time to control the motion of an autonomous mobile robot. This type of control (with some modifications) can be used in a lot of real world applications, like interoffice mail delivery, exploration of buildings in disaster areas, autonomous cars, etc.

The major enhancement of the system would be to use a better, faster microcontroller so that the fuzzy calculations could be done onboard rather than using a lookup table. Also it would be possible to add a few more ultrasonic sensors to allow it follow a corridor rather than a wall. Finally, the membership functions could be changed from triangular to other functions to have a more smooth control response.

REFERENCES

- [1] R. Frizera Vassallo, H.J. Schneebeli, J. Santos-Victor, Visual navigation: combining visual servoing and appearance based methods, International Symposium on Intelligent Robotic Systems, SIRS'98, Edinburgh, Scotland, July 1998.
- [2] J. Santos-Victor, G. Sandini, F. Curotto, S. Garibaldi, Divergent stereo in autonomous navigation: from bees to robots, International Journal of Computer Vision 14 (1995) 159-177.
- [3] R. Carelli, C. Soria, O. Nasisi, E. Freire, Stable AGV corridor navigation with fused vision-based control signals, Conference of the IEEE Industrial Electronics Society, IECON, Sevilla, Spain, November 5-8, 2002.
- [4] A. Dev, B. Kröse, F. Groen, Navigation of a mobile robot on the temporal development of the optic flow, IEEE/RSJ/GI International Conference on Intelligent Robots and Systems IROS'97, Grenoble, September 1997, pp. 558-563.
- [5] R. Benporad, M. Di Marco, A. Tesi, Wall-following controllers for sonar-based mobile robots, IEEE Conference on Decision and Control, San Diego, December 1997.
- [6] Ole Jakob Sordalen, Feedback Control of Nonholonomic Mobile Robots, doctoral thesis, Department of Engineering Cybernetics, The Norwegian Institute of Technology, 1993.
- [7] Danny Ratner, Phillip McKerrow, "Navigating an outdoor robot along continuous landmarks with ultrasonic sensing," Robotics and Autonomous Systems 45 (2003) 73-82.
- [8] Ricardo Carelli Eduardo Oliveira Freire, "Corridor navigation and wall-following stable control for sonar-based mobile robots," Robotics and Autonomous Systems 45 (2003) 235-247
- [9] Mohammad Jamshidi, et al. (eds.), Applications of fuzzy logic: Towards high machine intelligence quotient systems, Prentice-Hall, NJ, 1997.
- [10] <http://www.manufacturing.net/ctl/article/ca372359>
- [11] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338-353.
- [12] E.H. Mamdani, S. Assilian, Application of fuzzy algorithms for control of simple dynamic plant, Proceedings of the Institute of Electrical Engineers 121 (1974) 1585-1588.
- [13] M.I. Rieiro, P. Lima "Kinematics Models of Mobile Robots," <http://omni.isr.ist.utl.pt/~mir/cadeiras/robmove/Kinematics.pdf>
- [14] Dan Simon "Sum normal optimization of fuzzy membership functions," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10, no. 4 (2002) 363-384.