2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery

Fuzzy Logic Control for Wheeled Mobile Robots

Shengju SANG[1, 2]
1. School of Mechanical and Power Engineering,
East China University of Science and Technology,
Shanghai, P.R. China, 200237
e-mail: sang1108@163.com

Jichao ZHAO[2]
2. School of Information Science and Technology,
Taishan College
Shandong Taian, P.R. China, 271021
e-mail: name@xyz.com

Hao WU[1]
1. School of Mechanical and Power Engineering,
East China University of Science and Technology,
Shanghai, P.R. China, 200237
e-mail: wh@mail.ecust.edu.cn

Qi AN[1]
1. School of Mechanical and Power Engineering,
East China University of Science and Technology,
Shanghai, P.R. China, 200237
e-mail: anqi@ecust.edu.cn

***Abstract*—A new theoretical control method based on the dynamic behavior of wheeled robot is developed, and a mechanism of Fuzzy inference for designing a robust control system is present. The inputs are obtained from mounted ultrasonic sensors. A fuzzy logic controller is used to control the robot's motion along the predefined path. The robot control system is first modeled in Matlab Simulink and then the fuzzy logic rules are optimized for the best results possible. Experimental results are presented to show the performance of the controller.**

Keywords-Fuzzy Logic Control; Wheeled Mobile Robots; robotic behavior; Dynamic and kinematic model; Matlab

I. INTRODUCTION

Wheeled 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, through distance or proximity measurements. The most common sensors are distance sensors (ultrasonic, laser, and so on.) capable of detecting obstacles and of measuring the distance to walls close to the robot path. When advanced autonomous robots navigate within indoor environments, they have to be endowed with the ability to move through corridors, to follow walls, to turn corners and so on. This type of robot is widely applied in habitat mapping, environmental monitoring, mapping and surveillance, search and rescue operations, planetary surface exploration, manufacturing, and service, and so on.

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 and adopting a host of assumptions. The latter is a popular one which often enables the formulation of viable control laws. However, the laws are typically valid only for systems that comply with imposed assumptions. 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 been proved a convenient tool for handling real world uncertainty and knowledge representation.

As regards the corridor and wall-following navigation problem, some control algorithms based on artificial vision have been proposed. In [1], cameras are used to drive the robot along the corridor axis or follow a wall, by using optic flow computation and its temporal derivatives. In [2], image processing is used to detect perspective lines to guide the robot along the center axis of the corridor. In [3], a globally stable control algorithm for wall-following based on incremental encoders and one sonar sensor is developed. In [4], an ultrasonic sensor is used to steer an autonomous robot along a concrete path using its edged as a continuous landmark. References [5] deal with algorithms for the synthesis of a kinematic controller, and a novel algorithm with a dynamic fuzzy controller applied to control the trajectory of robots with two independent wheels is proposed.

In Section 2 the mathematical model of the robot is developed. In Section 3 the fuzzy logic controller is developed and discussed in detail. In Section 4 the hardware and software design of the robot are discussed. Performance comparisons are given to demonstrate the efficiency of the proposed method in Section 5. Finally the conclusions drawn from the results obtained are given in Section 6.

II. DYNAMIC AND KINEMATIC MODEL

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 paper is a wheeled robot intended for indoor use. This robot type is the easiest one to model, control, and build. There are various behaviors that could be modeled, like wall following, collision avoidance, corridor following, goal seeking, adaptive goal seeking.

978-0-7695-3735-1/09 $25.00 © 2009 IEEE
DOI 10.1109/FSKD.2009.241

IEEE computer society

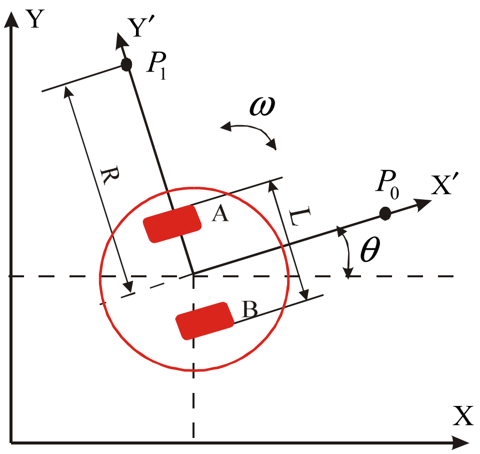

Figure 1. Kinematic model of the robot

Consider a generic mobile vehicle with two independent wheels. Its motion towards P0, which is governed by the combined action of both the angular velocity ω and the linear velocity v, is always on the direction of one of the axes of the frame attached to the robot. Fig. 1 shows the variables used in the kinematic and dynamic equations discussed below. "L" is distance between the two wheels. "R" is instantaneous curvature radius of the robot trajectory, relative to the mid-point of the wheel axis. P1 is instantaneous center of curvature.

A. *Governing Equations defining the robot*

The usual set of kinematics equations, which involves Cartesian position X-Y of the robot are as follows [6]:

$$\begin{aligned} vx(t) &= v(t)\cos\theta(t) \\ vy(t) &= v(t)\sin\theta(t) \end{aligned} \tag{1}$$

Therefore the linear distance of the robot is given as

$$\begin{aligned} x(t) &= \int_0^t v(t)\cos(\theta(t))dt \\ y(t) &= \int_0^t v(t)\sin(\theta(t))dt \\ \theta(t) &= \int_0^t \omega(t)dt \end{aligned} \tag{2}$$

The above equations can also be represented in the following from:

$$\begin{aligned} \begin{bmatrix} vx(t) \\ vy(t) \\ \omega(t) \end{bmatrix} &= \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos\theta \\ v(t)\sin\theta \\ \omega(t) \end{bmatrix} \\ &= \begin{bmatrix} (va+vb)\cos\theta/2 \\ (va+vb)\sin\theta/2 \\ (vb-va)/L \end{bmatrix} = \begin{bmatrix} \cos\theta/2 & 0 \\ \sin\theta/2 & 0 \\ -1/L & 1/L \end{bmatrix} \begin{bmatrix} vb(t) \\ va(t) \end{bmatrix} \end{aligned} \tag{3}$$

Where

vx (t) = linear velocity of the robot on the X direction in the world frame

vy (t) = linear velocity of the robot on the Y direction in the world frame

vq(t) = linear velocity of a wheel in the world frame

vb (t) = linear velocity of B wheel in the world frame.

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.

B. *Types of robotic behaviors and controllers*

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 (velocities or torques) to the mobile robots 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.

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. 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. 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 behavior of an autonomous mobile robot.

III. FUZZY LOGIC CONTROLLER (FLC)

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

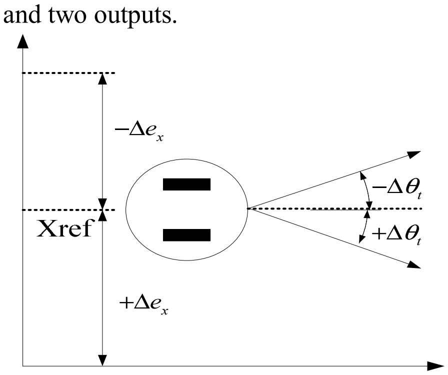

Figure 2. Robot in Cartesian space

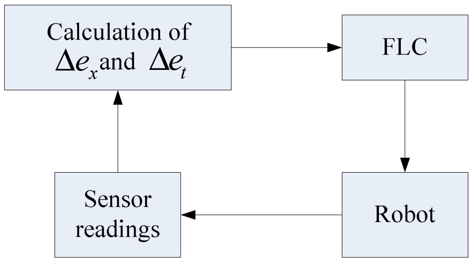

Figure 3. Block diagram of the robotic system

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 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.

A. Fuzzy inference system

In the fuzzy inference system designed in this system, the membership functions of the linguistic variables are sum normal triangular functions. It 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 fuzzy inputs and the areas of these functions is simpler and faster compare to other membership functions such as Gaussian. The rule base for the FLC proposed in this study is listed in Tables 1. 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.

In the TABLE 1, "N", "Z", "P" indicate negative, zero and positive respectively. While "S", "M", "F" present slow, medium and fast severally.

TABLE I. RULE BASE FOR ω

$\Delta e_x / \Delta e_t$	$\Delta\omega_B$			$\Delta\omega_A$		
	N	*Z*	*P*	*N*	*Z*	*P*
N	*S*	*S*	*S*	*F*	*M*	*S*
Z	*S*	*F*	*M*	*M*	*F*	*S*
P	*S*	*M*	*F*	*S*	*S*	*S*

B. Sum normal fuzzification

Consider the i[th] fuzzy membership function of the jth input zj. Its modal pint is denoted as cij, its lower half width as bij- , and its upper half width as bij+. The membership function attains a value of 1 when the input is cij. As the input decreases from cij, the membership function value decreases linearly to 0 at cij- bij- and remains at 0 for all inputs les than cij - bij-. As the input increases from cij, the membership function value decreases linearly to 0 at cij + bij+ and remains at 0 for all inputs greater than cij + bij+. The membership function of the j[th] crisp input zj in its ith fuzzy set is given by

$$f_{ij}(z_j) = \begin{cases} 1+(z_j - c_{ij})/b_{ij}^- \\ 1-(z_j - c_{ij})/b_{ij}^+ \\ 0 \end{cases} \quad (4)$$

The 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 fuzzy logic system (FLS). The fuzzy output is mapped to a crisp number y using centroid defuzzification. Thus, the final fuzzy output equations are derived mathematically, just as follows: [6, 7]

$$\hat{y} = (\sum_{j=1}^{M} w_j \Gamma_j J_j) / \sum_{j=1}^{M} w_j J_j) \quad (5)$$

Where w_j is the active level of the consequent of the jth rule, Γ_j and J_j are the centroid and the area of the jth output fuzzy membership function.

$$\Gamma_j = \frac{\beta_j^+(3\gamma_j + \beta_j^+) + \beta_j^-(3\gamma_j + \beta_j^-)}{3(\beta_j^+ + \beta_j^-)} \quad (6)$$

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

IV. DESIGN 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-501Hp manufactured by MPI and are rated at 4.8 VDC. They are run using PWM signals generated from the PIC16F877 microcontroller.

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, the response of the system in real time is too slow to allow the robot to attain a steady state. Thus have a lookup table is used instead of 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. Fig. 4 shows a block diagram of the hardware setup.

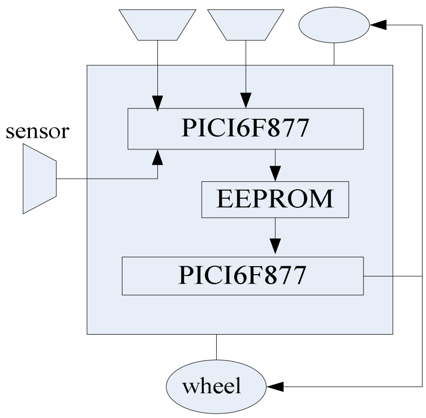

Figure 4. Block diagram of the hardware.

B. *Simulink model description*

The Simulink model is shown in Fig. 5. The fuzzy block in the Simulink model is a customized Matlab m-file function block replacing Matlab 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 Xref input is intended to keep the robot a certain distance away from the wall which it is following. The Tref 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.

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 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 steady state much earlier with a FLC than a P controller.

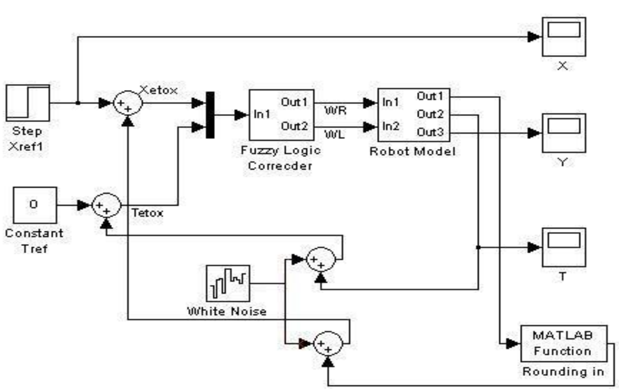

Figure 5. Simulink model of the robot.

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 nonholonomic wheeled mobile robot to test the performance of the proposed fuzzy controller. A Fuzzy Logic Controller has been implemented in real time to control the motion of an autonomous mobile robot. This type of control can be used in a lot of real world applications, like interoffice mail delivery, autonomous cars, and so on.

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. Finally, the membership functions could be changed from triangular to other functions to have a more smooth control response.

ACKNOWLEDGMENTS

The work was partially supported by the science and technology development fund of Tai'an (20082011) and the Foundation provided by Taishan College Y06-2-15).

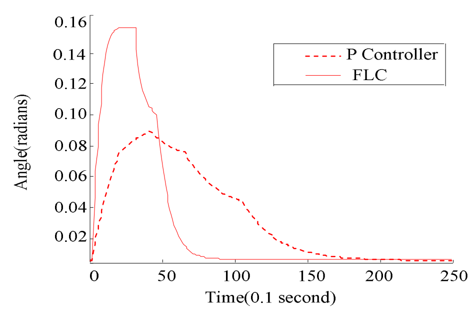

Figure 6. Angle of orientation as afuncion of time

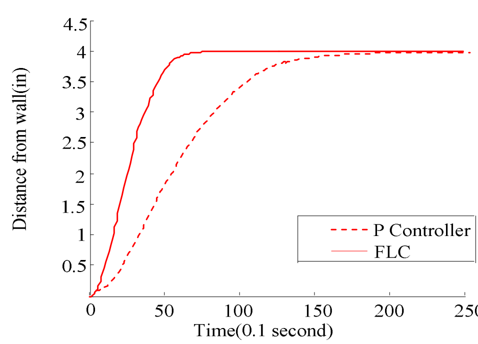

Figure 7. Distance from the wall as a function of tim

REFERENCES

[1] R. Carelli, C. Soria, and O. Nasisi, "Stable AGV corridor navigation with fused vision-based control signals", Conference of the IEEE Industrial Electronics Society, IECON, Sevilla, Spain, 3(2002), pp. 2433- 2438.

[2] R. Frizera Vassallo, H.J. Schneebeli, J. "Visual navigation: combining visual servoing and appearance based methods", International Symposium on Intelligent Robotic Systems, SIRS'98, Edinburgh, Scotland, 1998.

[3] R. Benporad, M.D. Marco, A.Tesi, "Wall-following controllers for sonar-based mobile robots", IEEE Conference on Decision and Control, San Diego, 36(1997), pp. 3063-3068.

[4] Danny Ratner, Phillip McKerrow, "Navigating an outdoor robot along continuous landmarks with ultrasonic sensing", Robotics and Autonomous Systems, 45 (2003), pp. 73-82.

[5] M.R. Francesco, M. Maurizio, "A new fuzzy robust dynamic controller for autonomous vehicles with nonholonomic constraints", Robotics and Autonomous Systems 52 (2005), pp. 115–131.

[6] V.M. Peri, Dan Simon, "Fuzzy logic control for an autonomous robot", Fuzzy Information Processing Society of Annual Meeting of the North American, 2005, pp. 337-342.

[7] Dan Simon, "Sum normal optimization of fuzzy membership functions", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 4 (2002), pp. 363-384.

[8] LIU Yong, PENG Zhenghong, "Design and simulation of fuzzy logic control system based on MATLAB", Engineering Journa1 of Wuhan University, vol. 41, no. 2 (2008) 132-135.

[9] ZHOU Pucheng, HAN Yusheng, XUE Mogen. "Multiple Robots Cooperative Pursuit Based on Motor Schema", Computer Engineering, vol. 341, no. 7 (2008) 32-3