Predictive Fuzzy Control for a Mobile Robot with Nonholonomic Constraints

Xianhua Jiang, Yuichi Motai, and Xingquan Zhu, *Member, IEEE*

***Abstract*— Automatic control of a mobile robot with nonholonomic constraints normally depends on complex signal processing mechanisms, e.g. predictive control. This paper presents a new trajectory tracking method for a mobile robot by combining predictive control and fuzzy control. To overcome the time delay caused by the slow response of the sensor, the algorithm employs predictive control to predict the position and orientation of the robot. In addition, fuzzy control is adopted to deal with nonlinear characteristic of the system. The advantages of this predictive fuzzy controller include high reliability for a slow sensor response, small error of absolute tracking, availability of a linearized predictive model and simplified fuzzy rules, which reduce the computing complexity. In our experiments, we applied this control method to the soccer robot. Accuracy and convergent performance was compared with a traditional PID controller, as well as with a conventional fuzzy controller. The experiment results demonstrated the feasibility and advantages of this predictive fuzzy control on the trajectory tracking of a mobile robot.**

***Index Terms*— autonomous navigation, mobile robot, predictive fuzzy control, trajectory tracking.**

I. Introduction

In recent years, the need to navigate automatic mobile robots has continued to increase. The navigation can occur in known or unknown environment. In the unknown environment, the most challenging task comes from the natural terrain. The typical solution is to acquire the environment information based on sensor signals and then use fuzzy control to avoid obstacles and reach the goal [1][2][3][8][15]. There are also other methods, such as the real-time sensor based navigation method using Kohonen's topology conserving network for navigation of a mobile robot in any uncertain environment proposed by I.J. Nagrath et al. [14].

In the known environment, the robot navigation can be divided into several steps. The first step involves predicting the trajectory of moving objects, which can utilize the mechanism proposed in [4], and may be used for path planning of a mobile robot. However, this existing solution doesn't address how to track moving objects. The second step involves navigating the mobile robots to avoid the obstacle based on acquired sensor signals. For example, Enrique J. et al. [6] computed the minimum distance between two mobile objects to predict and avoid collisions. The third step involves planning the path, which may utilize the mechanisms suggested in [5], where linear and angular maximum velocities, as well as dynamic constraints were considered. When it comes to trajectory tracking, Elnagar et al. [7] introduced a two-module fuzzy logic controller for autonomous navigation and control of small manned-unmanned aerial vehicles, but their solution didn't consider the time delay caused by the slow response of sensors, which is nontrivial in real-world applications.

In this paper, we mainly consider trajectory tracking of autonomous mobile robots, where mobile robots usually have nonlinear time-delay characteristics and are often perturbed by additive noise. For nonlinear problems, many existing experiments have demonstrated that a fuzzy controller has good performance in dealing with the additive noise. As a result, fuzzy control is usually applied to a complex system whose dynamic model is not well defined or not available at all. In addition to handling nonlinear problems, the fuzzy control can also enhance the robustness of whole robot system. However, when it comes to certain other situations, such as large delay, the control performance of the fuzzy controller is deteriorated. An alternative solution to these problems is to adopt a predictive control. A predictive control model can cope with the big delay and has good performance where a fuzzy controller is inferior.

From the above observations, we reach the conclusion that fuzzy control can cope with nonlinear characteristic of the system, while predictive control can deal with large time delay of sensor signals. A system framework that combines these two approaches appears to be promising for the real-world autonomous mobile robot tracking. Our experimental evaluations demonstrate the feasibility of this new strategy. The rest of this paper is organized as follows. We will first analyze the kinematical model of a mobile robot in Section II. In Section III, we'll discuss the design of the predictive controller and fuzzy controller. Experimental results are reported in Section IV to demonstrate the performance of the whole system. Conclusions are given in Section V.

Manuscript received November 28, 2004. This work was supported in part by University of Vermont Experimental Program to Stimulate Competitive Research Program (EPS-0236976).

Xianhua Jiang is with the Department of Electrical Engineering, University of Vermont, Burlington, VT 05405 USA.

Yuichi Motai is with the Department of Electrical Engineering, University of Vermont, Burlington, VT 05405 USA.

Xingquan Zhu is with the Department of Computer Science, University of Vermont, Burlington, VT 05405 USA.

II. KINEMATICAL MODEL OF MOBILE ROBOT

In this paper, we take a two-wheeled mobile robot as an object. The wheel rotation is limited to one axis, and the navigation is determined by the speed change in either side of the robot. Therefore, this kind of robot has nonholonomic constraints which should be considered during path planning. The kinematical scheme of a mobile robot can be depicted as Fig. 1, where V is the velocity of robot centroid, V_L is the velocity of the left wheel, V_R is the velocity of the right wheel, r is the radius of wheel, L is the distance between two wheels, x and y are the position of the mobile robot, and θ is the orientation of the robot.

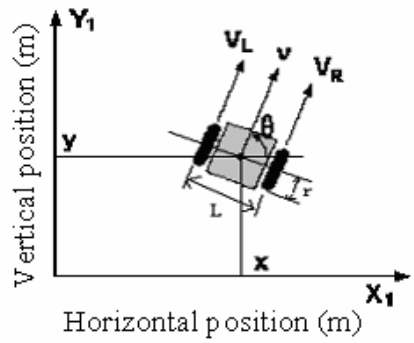

Fig. 1 Kinematical Scheme of the Mobile Robot

According to the motion principle of rigid body kinematics, the motion of a mobile robot can be presented as following (1) and (2), where ω_L and ω_R are angular velocities of left wheel and right wheel respectively, ω is the angular velocity of centroid.

$$V_R = r\omega_R \ , \quad V_L = r\omega_L \tag{1}$$

$$\omega = \frac{V_R - V_L}{L} \ , \quad V = \frac{V_R + V_L}{2} \tag{2}$$

Combining (1) with (2), we can obtain

$$\omega = \frac{r}{L}(\omega_R - \omega_L), \ V = \frac{r}{2}(\omega_R + \omega_L). \tag{3}$$

Moreover, the dynamic function of the robot centroid can be defined by (4).

$$\dot{x} = V\cos\theta, \quad \dot{y} = V\sin\theta \ , \quad \dot{\theta} = \omega \tag{4}$$

According to (3) and (4), we can get (5) and (6):

$$\omega_R = \frac{1}{r}V + \frac{L}{2r}\omega \ , \quad \omega_L = \frac{1}{r}V - \frac{L}{2r}\omega \tag{5}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix} \tag{6}$$

Equations (5) and (6) describe the kinematical model of a wheeled mobile robot. The controlled variables of the model are the position and orientation of the mobile robot, while the control variables are the angular velocities of the left wheel and the right wheel. We can also see that this is a nonlinear system. Time delay and noise will also be fed into the model during the acquisition of position and orientation. Therefore, in this paper, we apply the predictive fuzzy control to improve the control performance of auto navigation.

III. DESIGN OF PREDICTIVE FUZZY CONTROLLER

Predictive fuzzy control can be described as follows: First, give a set point, which is the goal of the mobile robot. According to the set point and the current robot position, a reference trajectory can be chosen. From these, the next reference position is determined. At the same time, the predictive controller can predict the next position of the robot using the current velocities of the left and right wheels. Fed into the difference between the next reference position and the predicted next position, the fuzzy controller can output the next angular velocities of the two wheels. The whole control system is shown as Fig. 2, where Y_d is the position and orientation of the set point; $Y(k)$ is the current position and orientation; $Y_r(k+1)$ is the next reference position and orientation; $Y_m(k+1)$ is the predicted next position and orientation; $Y_c(k+1)$ is the compensated predicted next position and orientation; E is the error between $Y_r(k+1)$ and $Y_c(k+1)$; $U(k)$ is the control variable, including the angular velocities of the two wheels.

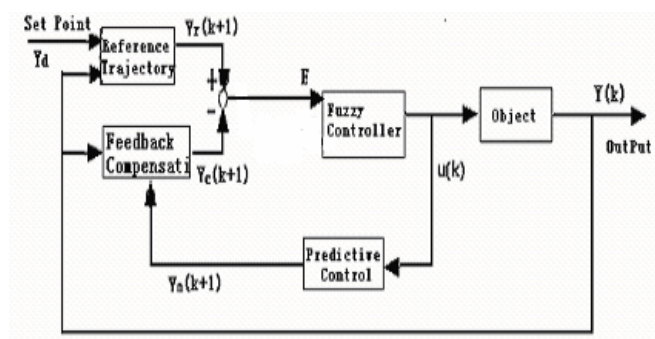

Fig. 2 Predictive Fuzzy Control System

In order to clarify the design, we have divided the predictive fuzzy controller into two parts: part A is the predictive control, and part B is the fuzzy control.

A. Predictive Control

Predictive control technology shows great advantages in delay and nonminimum phase systems, which are asymptotically stable without feedback. But, when it comes to nonlinear plants, predictive control technology has difficulty in keeping up with real time. The amount of computations needed to solve this problem online may make the approach inapplicable to a practical system. Therefore, we first need to linearize the system. Then we use this linearized model to

predict the next position and orientation of the mobile robot. A traditional predictive control is composed of a predictive model, feedback compensation, and online optimization. But this predictive fuzzy control system only needs a predictive model and feedback compensation; online optimization is substituted by the fuzzy controller. The design of the predictive controller consists of three steps: the development of a predictive model, the implementation of feedback compensation and the production of a reference trajectory.

Step1: Predictive model

In this step, the nonlinear model is linearized. According to (6), we have (9).

$$x_{k+1} - x_k = \int_{kT}^{(k+1)T} V\cos\theta dt .$$

$$y_{k+1} - y_k = \int_{kT}^{(k+1)T} V\sin\theta dt \qquad (7)$$

$$\theta_{k+1} - \theta_k = \int_{kT}^{(k+1)T} \omega dt$$

We predict the next position of robot according to the current velocities of the left and right wheels. So, we have

$$V = V_k , \quad \theta = \theta_k + \omega_k t . \qquad (8)$$

From (7) and (8), we can get the linearized predictive model in (9).

$$x_{k+1} = x_k + \frac{V_k}{\omega_k}\left(\sin\theta_{k+1} - \sin\theta_k\right)$$

$$y_{k+1} = y_k + \frac{V_k}{\omega_k}\left(\cos\theta_k - \cos\theta_{k+1}\right) \qquad (9)$$

$$\theta_{k+1} = \theta_k + \omega_k T$$

Therefore, according to k^{th} V and ω, get $(k+1)^{th}$ x, y and θ. Equation (9) represents the predictive model of our control system. In the next step, the predicted value is compensated.

Step 2: Feedback Compensation

Because of the model errors, nonlinear characteristics, disturbance etc., there are discrepancies (errors) between the predictive output and the actual output. In order to make predictive value more approximate, feedback control based on the predictive model is proposed to compensate the predictive output. With the error between predictive output and measured output, feedback control algorithm can be described as (10), where $Y_c(k+1)$ is the $(k+1)^{th}$ compensated predictive output; $Y_m(k+1)$ is the $(k+1)^{th}$ predictive output based on predictive model. $Y(k)$ is the kth measured output. $Y_c(k)$ is the kth compensated predictive output.

$$Y_c(k+1) = Y_m(k+1) + [Y(k) - Y_m(k)] \qquad (10)$$

After compensation, the predicted value is much more reliable than before. This value is then compared with the reference value. Step (3) describes the method for determining the reference value.

Step3: Reference Trajectory

Since the emphasis of this paper is trajectory tracking, we adopted a typical existing method to produce the reference trajectory, instead of designing our own algorithm. Nowadays, a lot of research has been done in this area, and many algorithms have been designed. These algorithms have considered obstacle avoidance, which makes algorithms applicable in the real environment. Y. Fukazawa et al. [9] expressed the working environment in grid pointes and regenerated the path using one that was planned beforehand. S. Fujisawa et al. [11] used reinforcement-learning systems and Cerebellar Model Articulation Controllers (CMACs), which is based on a trial-and-error search for a course plan. Zhu Yongjie et al. [12] applied neural network to the path planning algorithm. H. Bruyninckx et al. [13] used Pythagorean hodograph curves to robot path planning. There are also other solutions, such as the potential field algorithm and the graph-searching algorithm. Considering the nonholonomic constraints of our mobile robot, we adopted the method proposed by G. Yasuda et al. [10] which considered kinematical constraints in the steering control of a wheeled mobile robot and used a genetic algorithm to generate an obstacle-free path.

In this part A, the design procedure of the predictive control is introduced. The design procedure consists of creating a predictive model, implementing feedback compensation and determining a reference trajectory. In the next section, the design of fuzzy control is presented.

B. Fuzzy Controller

The structure of fuzzy control is shown in Fig. 3. The three inputs are position error E_x, E_y, and orientation error E_θ between the reference value and predicted value. The outputs are V and ω, where V is the velocity of the robot centroid, and ω is the angle velocity of the robot centroid.

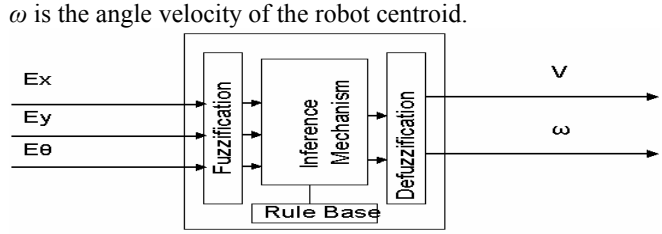

Fig. 3 Block Diagram of the Fuzzy Controller

This fuzzy controller has three inputs and two outputs; therefore it will produce a large scale of control rules. In order to make the computing time reasonable, we need to simplify the rules.

From the above Reference Trajectory section, we know that the nonholonomic constraints have been considered during path planning. Therefore, the trajectory of the mobile robot must be a smooth curve, and the orientation of the robot can't change suddenly. From (9), we know that if the $(k+1)^{th}$ predicted θ is equal to the $(k+1)^{th}$ reference θ , then we can't adjust $(k+1)^{th}$ ω, otherwise θ would change. In this case, the change of x and y is only caused by V. We define $E_d = E_x\cos\theta_{k+1} + E_y\sin\theta_{k+1}$. When $E_\theta = 0$, E_d has a definite physical meaning, which presents the displacement of the next

reference position from the next compensated predictive position. When E_d is larger than zero, V is increased. When E_d is smaller than zero, we need to decrease V. Therefore, the rough rules for fuzzy control are the following:

- If E_θ is larger than zero, then increase ω, keep V at the previous value.
- If E_θ is smaller than zero, then decrease ω, keep V at the previous value.
- If E_θ is zero, and E_d is larger than zero, then increase V while ω unchanged.
- If E_θ is zero, and E_d is smaller than zero, then decrease V while ω unchanged.
- Otherwise, keep ω and V at their previous values.

The fuzzy rules will be discussed in detail as follows. The design consists of three steps: fuzzification, fuzzy rules, and defuzzification.

Step 1: Fuzzification

In order to keep the number of fuzzy rules at a reasonable level, we define the fuzzy sets of inputs and outputs as the same as {PB (Positive Big), PM (Positive Middle), PS (Positive Small), ZE (Zero), NS (Negative Small), NM (Negative Middle), and NB (Negative Big)}.

After defining the fuzzy sets, our fuzzy controller utilizes symmetric triangular membership functions on the controller's inputs and outputs. The membership functions of inputs and outputs are shown in Fig.4, where $E_d \in [-20, +20]$ cm, $E_\theta \in [-\pi, +\pi]$, $V \in [-2, +2]$ m/s, $\omega \in [-2\pi, +2\pi]$ radiant/s. These small ranges can make the fuzzy controller sensitive to the small changes in position.

Let's take the membership function of E_d as an example. The function μ qualifies the certainty that E_d can be classified linguistically as Positive Big, Positive Middle et al. When E_d =0, μ =1 means that we are absolutely certain that E_d =0 is Zero. When E_d =3.33 cm, μ =0.5 indicates that we are half certain that E_d =0 is Zero. When E_d =6.66 cm, μ =0 means that we are certain that E_d =6.66 cm is not Zero. (Actually, it is Positive Small.)

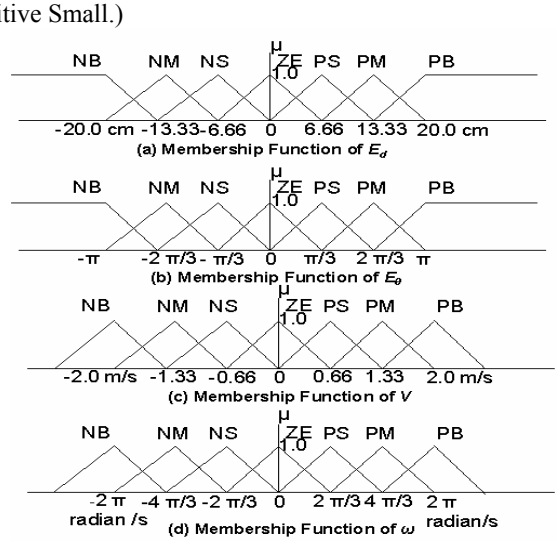

Fig.4 Membership Function

Step 2: Fuzzy Rules

In this step, we use the linguistic quantification to specify a set of rules that captures the expert's knowledge about how to control the plant. For example:

If E_θ is NB **Then** ω is NB, and V is ZE. This rule quantifies the position where the predicted angle of the mobile robot is much larger than the reference angle. Therefore, we must decrease ω to reduce the orientation of robot.

When E_θ is not ZE, we design the fuzzy rules as below:

- **If** E_θ is NM **Then** ω is NM, and V is ZE.
- **If** E_θ is NS **Then** ω is NS, and V is ZE.
- **If** E_θ is NB **Then** ω is NB, and V is ZE
- **If** E_θ is PS **Then** ω is PS, and V is ZE.
- **If** E_θ is PM **Then** ω is PM, and V is ZE.
- **If** E_θ is PB **Then** ω is PB, and V is ZE.

From the above linguistic rules, we know that when E_θ is not ZE, we change V to adjust E_θ first, and we don't concern about E_d. Only after we have constrained E_θ to ZE, we will change V to adjust E_d. If E_θ is ZE Then ω is ZE, adjust V according to E_d. In this case, the complete set of rules is shown as below:

- **If** E_d is NM **Then** V is NM, and ω is ZE.
- **If** E_d is NS **Then** V is NS, and ω is ZE.
- **If** E_d is NB **Then** V is NB, and ω is ZE
- **If** E_d is PS **Then** V is PS, and ω is ZE.
- **If** E_d is PM **Then** V is PM, and ω is ZE.
- **If** E_d is PB **Then** V is PB, and ω is ZE.

In this step, the details of fuzzy rules have been explained. After designing the rules according to the value of E_θ, the number of rules is reduced to 6+6=12 from 7*7*7*2=686. Therefore, the complexity of the fuzzy system is decreased dramatically. The last step for the fuzzy controller is defuzzification.

Step 3: Defuzzification

Considering the real-time characteristics and the complexity of the algorithm, we use the max criterion to defuzzify the output variable. At first, we choose the rule for output which best fits the current situation. In other words, we are more certain that this rule can be applied than the other rules. Then, we find the output value that has the maximum membership function value according to this rule. If there is more than one variable that has the same maximum membership function value, the average of these variables is produced.

IV. EXPERIMENTAL RESULTS

The experiment is made on the soccer robot, which is an ideal testing paradigm. The platform is shown in Fig. 5. Micro-robot soccer is a game similar to classical soccer, where the only difference is that all the players are robots. Each team consists of three players, one of which is a goalkeeper. The individual robots don't possess intelligence. They are wirelessly controlled by the host computer, which is responsible for the game strategy. The color camera connected to the computer is used for gaining the position information of individual players and an orange golf ball (representing the

"soccer ball"). Therefore, this test platform has the nonholonomic constraints, stores the requirement on start and end position and orientation, and monitors the full environment information. Those characteristics are the premise of our control strategy.

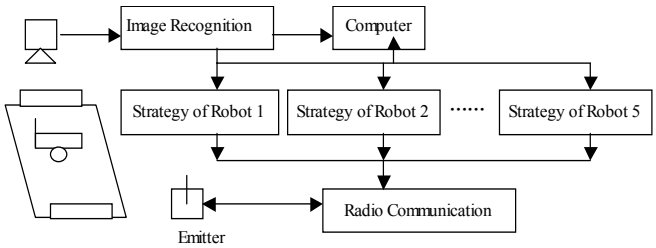

Fig.5 Soccer Robot Test Platform

Since we are investigating trajectory tracking, we only need a micro-robot. First, we set a start position and orientation as well as an end position and orientation. The computer produces a smooth curve from current position to the end position. Our sole concern is whether our robot can track this trajectory well; optimization of the curve is not addressed.

In this paper, we made two experiments to test the accuracy and convergent efficiency of our method respectively. The results are compared with traditional PID and fuzzy control.

Experiment 1: Tracking Error vs. Control Methods

In this experiment, we compare the accuracy of the tracking system as they are controlled by one of three methods: 1) a traditional PID controller, 2) a conventional fuzzy controller, and 3) our predictive fuzzy control. The traditional PID controller uses the distance and orientation error between the current point and the goal point as inputs. The outputs are the angle velocities of the left and right wheels. The conventional fuzzy controller uses the position error and orientation error between the measured value and the reference value as inputs.

We set the position of start point as (0, 0) meter and the orientation as 0 degree. The end position is set as (2, 2) meter, with the end orientation as 90 degree. For each method we perform two experiments, in which the response time of the sensor is controlled in 150 ms and 800 ms respectively. Here, the sensor response time is calculated from image acquisition till computing out the position and orientation of the robot.

Fig.6 (a) –(c) shows the tracking errors controlled by the traditional PID controller, the conventional fuzzy controller and the predictive fuzzy controller. The tracking error is the absolute distance between the reference position and the measured position, instead of the distance between current position and the goal position. The dash curve denotes the tracking error with sensor response time as 800ms. Another curve represents the tracking error with sensor response time as 150ms.

From Fig.6, it is evident that when the sensor has a quick response, there is no much difference between these three strategies. But for a slow sensor response, the robot controlled by the traditional PID controller becomes unstable as shown in Fig.6 (a). The convergence rate of this problem has been improved by conventional fuzzy control in Fig. 6(b). The influence by different sensor response time is obviously reduced by the predictive fuzzy control. As for error variation, the range for the traditional PID controller appears to be about 0~18 cm. The range for conventional fuzzy control has been decreased to about 0~12 cm, while the predictive fuzzy controller has the smallest range about 0~8 cm. Therefore, when the sensor has a slow response time, the system using the predictive fuzzy controller has the best reliability.

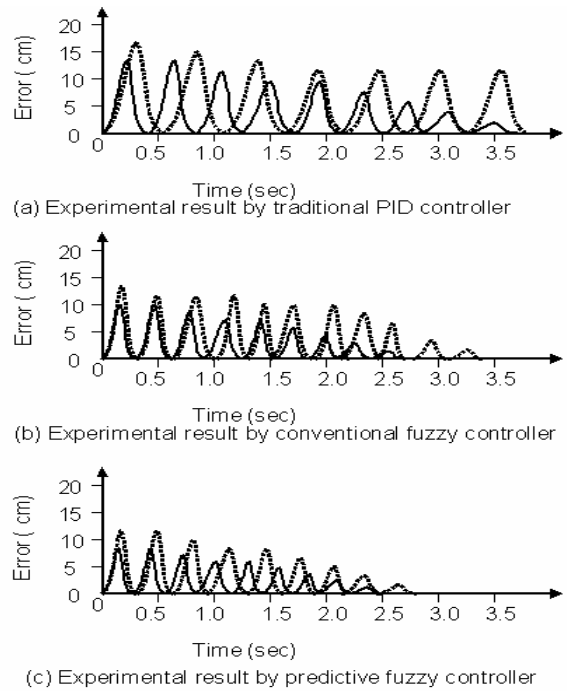

Fig.6 Tracking Error by three Controllers

Experiment 2: Navigation Time vs. Different Goal Position

In this experiment, we compare the convergent efficiency on different goal positions by using a traditional PID controller, a conventional fuzzy control and a predictive fuzzy controller. We set different goal positions and measured the navigation time of soccer robot with different control strategies. We described the position of the mobile robot in a 2D coordinate with units as (meter, meter). These different goals were (1, 1), (1.5, 1.5), (2, 2), (2.5, 2.5), (3, 3), and (3.5, 3.5) meter, with the same end orientation as 90 degrees. The common start point was (0, 0) and the orientation was 0 degrees. The sensor response time was 150 ms.

Fig.7 shows the measured navigation time of the three control strategies. The X-axis represents the end positions of the robot, while the Y-axis denotes the navigation time needed to reach the end position. There are three curves, which represent the traditional PID controller, the conventional fuzzy control and the predictive fuzzy controller respectively. For each kind of control strategy, we drew the curve by connecting the navigation time needed to reach the different end positions. These curves illustrate the navigation time

needed from the start position to the end position by using these control strategies.

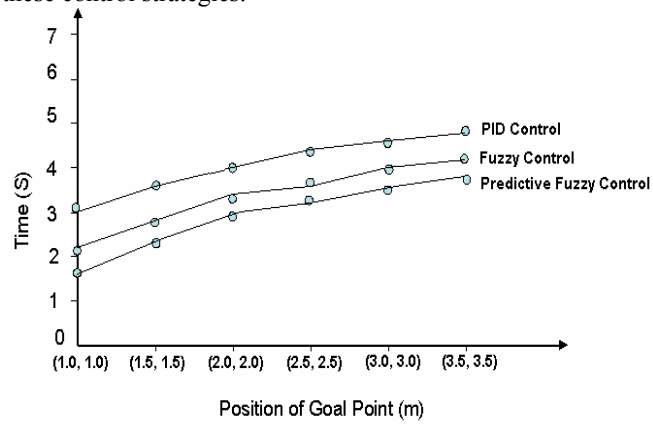

Fig.7 Navigation Time of Different Goal Point by three Controllers

From Fig.7, we can see the increasing tendency of navigation time with the increasing navigation distance. At the same time, we can evaluate the convergent efficiency of predictive fuzzy control by comparing with other two strategies. For example, when the end position is (1.0, 1.0), the navigation time of the traditional PID controller is 3.1 seconds, conventional fuzzy control is 2.1 seconds, and the predictive fuzzy control is 1.6 seconds. We compare the convergent performance of predictive fuzzy control with traditional PID control by calculating (3.1-1.6)/3.1=48.4%. Table I illustrates the convergent performance of predictive fuzzy control compared with other two strategies, where PF denotes predictive fuzzy control, PID denotes traditional PID control, and F denotes fuzzy control.

TABLE I
CONVERGENT PERFORMANCE OF PREDICTIVE FUZZY CONTROL COMPARED WITH OTHER TWO STRATEGIES

	Goal Positions (meter, meter)					
	(1.0,1.0)	(1.5,1.5)	(2.0,2.0)	(2.5,2.5)	(3.0,3.0)	(3.5,3.5)
PF vs. PID	45.2%	37.1%	30.3%	28.7%	26.7%	23.4%
PF vs. F	19.0%	15.4%	14.3%	11.4%	10.8%	10.0%

From Table I, we can see that, given different goal positions, predictive fuzzy control keeps the shortest navigation time. Especially when the goal is near the start points, traditional PID control and conventional fuzzy control require much more time to reach the target than predictive fuzzy control. The experiment results show that the predictive fuzzy control always has best performance among the three controllers tested regardless of whether the goal is near or far from the starting point.

V. CONCLUSION

In this paper, the predictive control theory has been incorporated into the fuzzy control to form a look ahead fuzzy logic control system, with the prerequisites that all environment information during the trajectory tracking are available and the robot has noholonomic constraints. The main advantages of this predictive fuzzy controller include high reliability for the slow sensor response, small error in absolute tracking, the availability of a linearized predictive model and simplified fuzzy rules which reduce the computing complexity. In conclusion, the proposed framework efficiently overcomes delay and nonlinear characteristics of the system and improves the robustness of a traditional fuzzy controller at the same time. The experiment results demonstrated the feasibility and advantages of this predictive fuzzy control on the trajectory tracking of mobile robots.

REFERENCES

[1] S. Boonphoapichart, S. Komada, T. Hori and W.A. Gruver, "Robot motion decision-making system in unknown environments," in *Proc. of 2003 IEEE Int. Conf. Robotics and Automation,*, vol. 3, pp. 4197-4202.

[2] L. Doitsidis,K.P. Valavanis and N.C. Tsourveloudis, "Fuzzy logic based autonomous skid steering vehicle navigation," in *Proc. 2002 IEEE Int. Conf. Robotics and Automation,*, vol. 2, pp. 2171-2177.

[3] Ayanna Howard, Homayoun Seraji and Edward Tunstel, " A Rule-Based Fuzzy Traversability Index for Mobile Robot Navigation," in *Proc. 2001 IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, vol.3, pp.3067-3072.

[4] Anmin Zhu and Simon X. Yang, "A Fuzzy Logic Approach to Reactive Navigation of Behavior-based Mobile Robots," in *Proc. 2004 IEEE/ASME Int. Conf. Robotics and Automation*, vol.5, pp. 5045-5050.

[5] G.Antonelli, S. Chiaverini and G. Fusco, "Experiments of fuzzy real-time path planning for unicycle-like mobile robots under kinematic constraints," in *Proc. 2002 IEEE Int. Conf. on Robotics and Automation,*, vol. 2, pp. 2147-2152.

[6] Enrique J.Bernabeu, Josep Tornero, and Masayoshi Tomizuka, "Collision Prediction and Avoidance Amidst Moving Objects for Trajectory Planning Applications," in *Proc. 2001 IEEE Int. Conf. Robotics and Automation,*, vol. 4, pp. 3801-3806.

[7] A.Elnagar and A.M. Hussein, "An adaptive motion prediction model for trajectory planner systems," in *Proc. 2003 IEEE/ASME Int. Conf. Robotics and Automation*, vol. 2, pp. 2442-2447

[8] H. Seraji, A. Howard and E. Tuustel, "Safe navigation on hazardous terrain," in *Proc. 2001 IEEE Int. Conf. Robotics and Automation,* vol. 3, pp. 3084-3091.

[9] Y. Fukazawa, T. Chomchana, J. Ota, H. Yuasa, T. Arai, and H. Asama, "Region exploration path planning for a mobile robot expressing working environment by grid points," in *Proc. 2003 IEEE Int. Conf. Robotics and Automation,*, vol. 2, pp. 2448-2454

[10] G. Yasuda and H. Takai, "Sensor-based path planning and intelligent steering control of nonholonomic mobile robots," in *27th Annual Conf. IEEE Industrial Electronics Society*, vol. 1, pp. 317-322, Nov. 2001

[11] S. Fujisawa, R. Kurozumi, T. Yamamoto and Y. Suita, "Path planning for mobile robots using an improved reinforcement learning scheme", in *Proc. 2002 IEEE Int. Symposium Intelligent Control*, pp. 67-74.

[12] Zhu Yongjie, Chang Jiang and Wang Shuguo, "A new path-planning algorithm for mobile robot based on neural network," in Proc. 2002 IEEE Region 10 Conf. Computers, Communications, Control and Power Engineering, vol. 3, pp. 1570-1573.

[13] H. Bruyninckx, and D. Reynaerts, "Path planning for mobile and hyper-redundant robots using Pythagorean hodograph curves," in Proc. 8th Int. Conf. Advanced Robotics, pp. 595-600, July. 1997

[14] I.J. Nagrath, L. Behera, K.M. Krishna, and K.D. Rajasekar, "Real-time navigation of a mobile robot using Kohonen's topology conserving neural network," in Proc. 8th Int. Conf. Advanced Robotics, pp. 459-464, July. 1997

[15] W.L. Xu, S.K. Tso, and Y.H. Fung, "Sensor-based reactive navigation of a mobile robot through local target switching," in Proc. 8th Int. Conf. Advanced Robotics, pp. 361-366, July. 1997