SMCia/05
2005 IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications
Helsinki University of Technology, Espoo, Finland, June 28-30, 2005

PREDICTIVE FUZZY LOGIC CONTROLLER FOR TRAJECTORY TRACKING OF A MOBILE ROBOT

Xianhua Jiang
Department of Electrical Engineering
University of Vermont
Burlington, VT 05405, USA
xjiang@emba.uvm.edu

Yuichi Motai
Department of Electrical Engineering
University of Vermont
Burlington, VT 05405, USA
ymotai@emba.uvm.edu

Xingquan Zhu
Department of Computer Science
University of Vermont
Burlington, VT 05405, USA
xqzhu@emba.uvm.edu

Abstract - This paper presents a new tracking method for a mobile robot by combining predictive control and fuzzy logic control. Trajectory tracking of autonomous mobile robots usually has non-linear time-varying characteristics and is often perturbed by additive noise. To overcome the time delay caused by the slow response of the sensor, the algorithm uses predictive control, which predicts the position and orientation of the robot. In addition, fuzzy control is used to deal with the non-linear characteristics of the system. Experimental results demonstrate the feasibility and advantages of this predictive fuzzy control on the trajectory tracking of a mobile robot.

I. INTRODUCTION

Navigating autonomous mobile robots has received considerable attention in recent years. In a known environment, robot navigation can be divided into several steps. The first step is to predict the trajectory of the moving objects, which can utilize the mechanism proposed in [1], and may be used for path planning of the mobile robot. Unfortunately, this existing solution, for the first step, does not consider how to track the moving objects. The second step involves navigating the mobile robots to avoid an obstacle based on acquired sensor signals. For example, Enrique J. et al. [3] computed the minimum distance between two mobile objects to predict and avoid collisions. The third step includes planning the path, which can utilize the mechanisms suggested in [2], where linear and angular maximum velocities, as well as dynamic constraints were considered. When it comes to trajectory tracking, Elnagar et al. [4] introduced a two-module fuzzy logic controller for autonomous navigation and control of small manned, as well as unmanned, aerial vehicles. Unfortunately, their solution did not consider the time delay caused by the slow response of sensors, which is a non-trivial problem in real-world applications.

With regard to trajectory tracking, mobile robots usually have non-linear time-varying characteristics and are often perturbed by additive noise. For non-linear problems, a fuzzy controller performs well in many existing experiments. In addition, fuzzy controllers can also enhance the robustness of the entire robot system. However, the performance of the fuzzy controller degrades when the system has a large delay. One solution to this problem is to use predictive control. A predictive control model can cope with the big delay and improve the tracking performance.

From the observations stated thus far, we reached the conclusion that a system framework combining fuzzy control with predictive control appears to be promising for the tracking of autonomous mobile robots. We will continue our discussion of this topic with three sub-topics, which follow this introduction. Section II analyzes the kinematical model of a mobile robot. Section III discusses the design of the predictive controller and the fuzzy controller. Lastly, experiments in Section IV demonstrate the performance of the entire system.

II. KINEMATICAL MODEL OF MOBILE ROBOT

A two-wheeled mobile robot was chosen as the object in this paper. Its wheel rotation is limited to one axis. Therefore, the navigation is controlled by the speed change on either side of the robot. This kind of robot has non-holonomic constraints, which should be considered during path planning. The kinematical scheme of a mobile robot can be depicted as in Fig. 1, where V is the velocity of the 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 each 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.

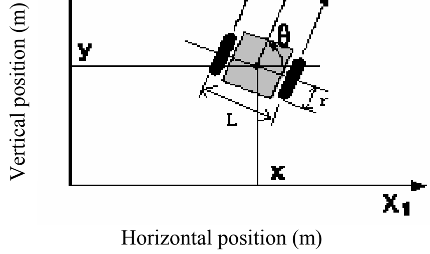

Fig. 1 Kinematical Scheme of the Mobile Robot

0-7803-8942-5/05/$20.00 @2005 IEEE.

According to the motion principle of rigid body kinematics, the motion of a mobile robot can be described using equations (1) and (2), where ω_L and ω_R are angular velocities of the left and right wheels respectively, and ω is the angular velocity of the 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

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

Moreover, we can define the dynamic function of the robot as (4).

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

Deriving (5) and (6) from (3) and (4):

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

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

Equations (5) and (6) describe the kinematical model of a two-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 non-linear system. Time delay and noise will also be fed into the model during the acquisition of position and orientation by a digital camera. Therefore, in this paper, we utilized predictive fuzzy control to improve the control performance of mobile robot navigation

III. Design of Predictive Fuzzy Controller

The complete predictive fuzzy control system is shown as Fig. 2. First, given a set point and the current position of the robot, a reference trajectory can be designed. From the reference trajectory, the next reference position is obtained. Meanwhile, the predictive controller predicts the next position of the robot using the current velocities of the left and right wheels. Based on the difference between the next reference position and the predicted position, the fuzzy controller can determine the next set of values for the angular velocity of each of the two wheels (left and right).

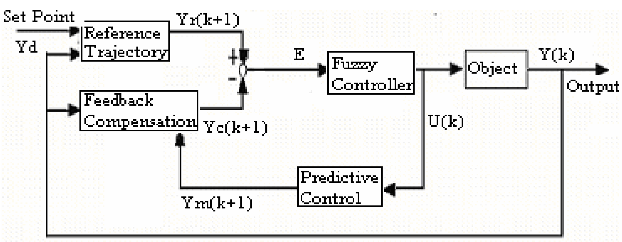

Fig. 2 Predictive Fuzzy Control System

Where Y_d is the pose of the set point, which includes the position and orientation the mobile robot; *Y(k)* is the current pose; $Y_r(k+1)$ is the next reference pose; $Y_m(k+1)$ is the predicted next pose. $Y_c(k+1)$ is the compensated predicted next pose. *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.

With the purpose of clarifying each aspect of the design, we have divided the predictive fuzzy controller into part A: *Predictive Controller*, and part B: *Fuzzy Controller*.

A. Predictive Controller

To non-linear applications, the computation of predictive control is very complicated. This makes it difficult to keep up with real time. Therefore, we first need to linearize the system. Then we use this linearized model to predict the next pose of the mobile robot. A traditional predictive control consists of a predictive model, feedback compensation, and online optimization. But here, we only need the predictive model and feedback compensation; online optimization is replaced by the fuzzy controller. The design of the predictive controller includes three steps: the development of a predictive model, the implementation of feedback compensation, and the production of a reference trajectory.

Step 1: Predictive model

In this step, the non-linear model is linearized. According to (6), we have

$$\begin{aligned} x_{k+1} - x_k &= \int_{kT}^{(k+1)T} V\cos\theta dt \cdot \\ y_{k+1} - y_k &= \int_{kT}^{(k+1)T} V\sin\theta dt \\ \theta_{k+1} - \theta_k &= \int_{kT}^{(k+1)T} \omega dt \end{aligned} \tag{7}$$

We predict the next position of robot in accordance with the current velocities of the left and right wheels. So, we have

$$V = V_k \ , \ \theta = \theta_k + \omega_k t \cdot \tag{8}$$

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

$$\begin{aligned} x_{k+1} &= x_k + \frac{V_k}{\omega_k}(\sin\theta_{k+1} - \sin\theta_k) \\ y_{k+1} &= y_k + \frac{V_k}{\omega_k}(\cos\theta_k - \cos\theta_{k+1}) \\ \theta_{k+1} &= \theta_k + \omega_k T \end{aligned} \tag{9}$$

Therefore, according to the k^{th} V and ω, we can obtain $(k+1)^{th}$ x, y and θ. Equation (9) depicts the predictive model of our control system. In the next step, the compensation of the predicted value occurs.

Step 2: Feedback Compensation

Due to model errors, non-linear characteristics, disturbance etc., there are errors between the predictive output and the actual output. In order to make the predictive value more accurate, we use feedback control to compensate for the error using (10).

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

After compensation, the reliability of the predicted value improves. Step 3 describes the methods for determining the reference value.

Step 3: Reference Trajectory

Since this paper emphasizes trajectory tracking, we adopted a typical existing method to produce the reference trajectory, instead of designing our own algorithm. Various reference trajectory algorithms have been developed recently, such as the potential field algorithm, graph-searching algorithm, reinforcement-learning algorithm, and genetic algorithm [5][6][7].

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

B. Fuzzy Controller

The fuzzy controller is shown in Fig. 3. The three inputs are pose error E_x, E_y, and E_θ between the reference value and predicted value. The outputs are V and ω.

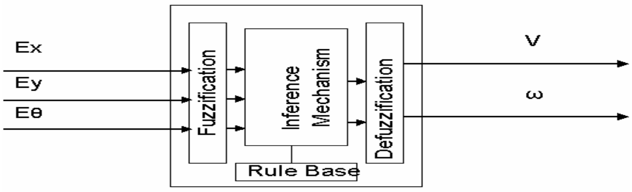

Fig. 3 Block Diagram of the Fuzzy Controller

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.

Since the trajectory of the mobile robot is a smooth curve, 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 represents 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.

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

Step 1: Fuzzification

To maintain fuzzy rules in a practical way, we define the fuzzy sets of inputs and outputs as {PB (Positive Big), PM (Positive Middle), PS (Positive Small), ZE (Zero), NS (Negative Small), NM (Negative Middle), and NB (Negative Big)}.

We perform symmetric triangular membership functions on the controller's input and output universes of discourse. As shown in Fig.4, the universe of discourse for the E_d is (-20, +20) cm, and for the E_θ (-π, + π). These small ranges can make the fuzzy controller sensitive to small changes in position. The universe of discourse for the V is (-2, +2) m/s, and for the ω (-2π, +2π) radiant/s.

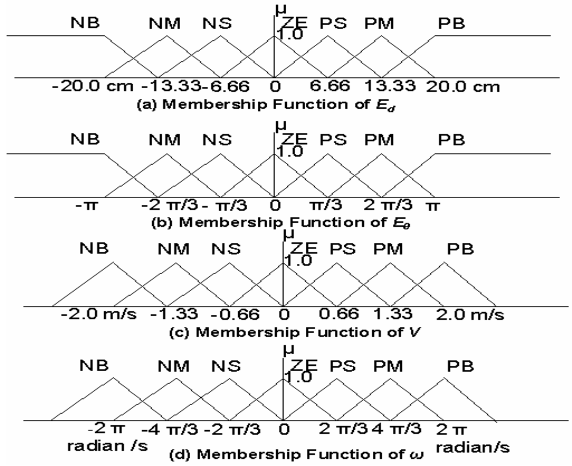

Fig.4 Membership Function

Step 2: Fuzzy Rules

In this step, we use the linguistic quantification to specify a set of rules that describe the expert's knowledge about how to control the objective. The fuzzy rules are shown in Table I.

TABLE I
FUZZY RULES(/ MEANS ANY VALUE)

E_θ	E_d	ω	V	E_θ	E_d	ω	V
NM	/	NM	ZE	ZE	NM	ZE	NM
NS	/	NS	ZE	ZE	NS	ZE	NS
NB	/	NB	ZE	ZE	NB	ZE	NB
PS	/	PS	ZE	ZE	PS	ZE	PS
PM	/	PM	ZE	ZE	PM	ZE	PM
PB	/	PB	ZE	ZE	PB	ZE	PB

From Table I, we know that when E_θ is not ZE (zero), 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 way, the number of rules is reduced to 6+6=12 from 7*7*7*2=686. Thus, the complexity of the fuzzy system decreases dramatically. At last, the fuzzy controller is defuzzified.

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. 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 used.

IV. Experimental Results

The experiment is made on the soccer robot used for a game similar to classical soccer. The individual robots 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 of individual players and an orange golf ball (representing the "soccer ball").

In this paper, we compare the accuracy of the tracking system by using three methods: 1) a traditional PID controller, 2) a conventional fuzzy controller, and 3) our predictive fuzzy control. The traditional PID controller uses the pose error between the current point and the goal point as inputs. The output is the angle velocities of the left and right wheels. The conventional fuzzy controller uses the pose error between the measured value and the reference value as inputs.

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 to the computation of the position and orientation of the robot.

Fig.5 (a)-(c) illustrates the tracking errors controlled by three methods. The tracking error is the absolute distance between the reference position and the measured position. The dash curve denotes the tracking error with sensor response time as 800ms, while the solid one represents the tracking error with sensor response time as 150ms.

Fig.5 clearly shows that when the sensor has a quick response, there is not 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.5 (a). The convergence rate of this problem has been improved by conventional fuzzy control in Fig. 5(b). The influence by different sensor response time is 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.

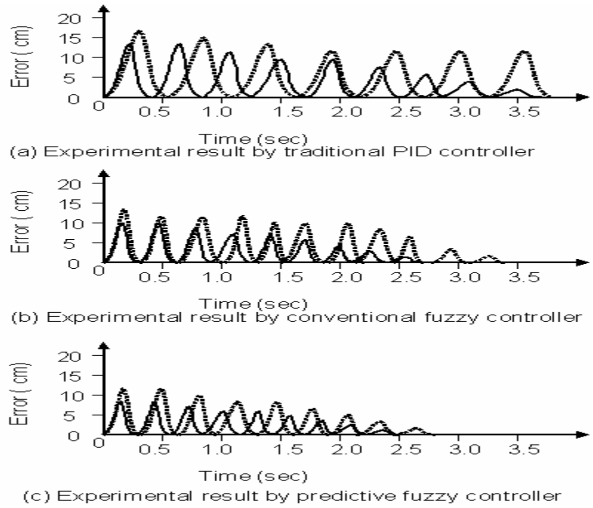

Fig.5 Tracking Error by three Controllers

V. Conclusion

This paper incorporates the predictive control theory into the fuzzy control to form a look ahead fuzzy logic control system, with the prerequisites that all environment information are available and the robot has non-holonomic constraints. The proposed framework efficiently overcomes delay and non-linear 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] Anmin Zhu and Simon X. Yang, "A Fuzzy Logic Approach to Reactive Navigation of Behavior-based Mobile Robots," *Proceedings of the 2004 IEEE/ASME International Conference on Robotics and Automation*, vol.5, pp. 5045-5050, April 2004

[2] G.Antonelli, S. Chiaverini and G. Fusco, "Experiments of fuzzy real-time path planning for unicycle-like mobile robots under kinematic constraints," *Proceedings of the 2002 IEEE International Conference on Robotics and Automation,*, vol. 2, pp. 2147-2152, May 2002

[3] Enrique J.Bernabeu, Josep Tornero, and Masayoshi Tomizuka, "Collision Prediction and Avoidance Amidst Moving Objects for Trajectory Planning Applications," *Proceedings of the 2001 IEEE International Conference on Robotics and Automation,*, vol. 4, pp. 3801-3806, 2001

[4] A.Elnagar and A.M. Hussein, "An adaptive motion prediction model for trajectory planner systems," *Proceedings of the 2003 IEEE/ASME International Conference on Robotics and Automation*, vol. 2, pp. 2442-2447, Sept 2003

[5] H. Seraji, A. Howard and E. Tuustel, "Safe navigation on hazardous terrain," *Proceedings of the 2001 IEEE International Conference on Robotics and Automation,* vol. 3, pp. 3084-3091, 2001.

[6] 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," *Proceedings of the 2003 IEEE International Conference on Robotics and Automation,*, vol. 2, pp. 2448-2454, Sept. 2003

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