

Research Article

A Two-Wheeled Self-Balancing Robot with the Fuzzy PD Control Method

Junfeng Wu,¹ Wanying Zhang,² and Shengda Wang¹

¹ College of Automation, Harbin University of Science and Technology, Harbin 150080, China

² Qingdao Haier AIC Elec Corp., Ltd., Qingdao Haier Co., Ltd., Qingdao 266101, China

Correspondence should be addressed to Wanying Zhang, zhangwy8@163.com

Received 19 September 2012; Revised 27 October 2012; Accepted 30 October 2012

Academic Editor: Peng Shi

Copyright © 2012 Junfeng Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A two-wheeled self-balancing robot with a fuzzy PD control method is described and analyzed as an example of a high-order, multiple-variable, nonlinear, strong-coupling, and unstable system. Based on a system structure model, a kinetic equation is constructed using Newtonian dynamics and mechanics. After a number of simulation experiments, we get the best Q , R , and state-feedback matrices. Then a fuzzy PD controller is designed for which the position and speed of the robot are inputs and for which the angle and angle rate of the robot are controlled by a PD controller. Finally, this paper describes a real-time control platform for the two-wheeled self-balancing robot that controls the robot effectively, after some parameter debugging. The result indicates that the fuzzy PD control algorithm can successfully achieve self-balanced control of the two-wheeled robot and prevent the robot from falling.

1. Introduction

Existing research on fuzzy reasoning can be roughly divided into three overlapping categories [1]: fuzzy reasoning methods and their analysis, logical foundations of fuzzy reasoning, and applications of fuzzy reasoning. Various fuzzy methods have been proposed based mainly on three different ideas. The first idea, composition, leads to the Zadeh compositional rule of inference (CRI) method [2, 3] and its variants [4–6]. The second idea is that of analogy and similarity. The third idea, interpolation analysis of fuzzy reasoning methods, is concerned with various properties of interest, such as interpretability of fuzzy rules, consistency of new fuzzy consequences with existing fuzzy premises, and continuity of fuzzy consequences with respect to fuzzy premises and fuzzy relations [7]. Numerous different implication operators and connectives can be adopted in fuzzy reasoning methods, including an important class of methods concerned with the suitability of particular fuzzy

reasoning methods for domain-specific applications [8]. The logical foundation of fuzzy reasoning is concerned with formal systems in which various fuzzy reasoning methods can be interpreted. Several formal systems of this kind have been proposed, including basic logic, monoidal t-norm-based logic, the quasiformal deductive system of Wang [9], and possibilistic logic [10]. These formal systems are mainly generalizations of classical propositional logic and multivalued logic. On the other hand, applications of fuzzy reasoning methods are extensive and can be found in disparate areas such as complex systems modeling and control [11], pattern recognition, decision-making [12], and safety monitoring.

In recent decades, motion control of robot manipulators has received a lot of attention [13]. Motion control is needed to make each joint track a desired trajectory as closely as possible. Many control algorithms, such as the computer torque method [14], optimal control [15], adaptive control [16], variable structure control (VSC) [17], neural networks (NNs), and fuzzy systems [18], have been proposed to deal with this robotic control problem. In [19, 20], a computer torque control is developed on the basis of feedback linearization. However, these designs are possible only when the robotic dynamics are well known.

A fuzzy logic controller (FLC) makes control decisions using well-known fuzzy IF-THEN rules. FLCs can be classified into two major categories: the Mamdani-type FLC, which uses fuzzy numbers to make decisions [21], and the Takagi-Sugeno (TS)-type FLC, which generates control actions using linear functions of the input variables. In the early years, most FLCs were designed by trial and error. Since the complexity of an FLC will increase exponentially when it is used to control complex systems, it is tedious to design and tune FLCs manually for most industrial problems such as robotic systems. This is why conventional nonlinear design methods [22] such as fuzzy sliding control, fuzzy gain scheduling [8], and adaptive fuzzy control [17] were adopted in the fuzzy control field to alleviate difficulties in constructing the fuzzy rule base.

Analytical calculations show that a two-input FLC employing a proportional error signal and a velocity error signal is a nonlinear proportional-integral (PI) or proportional-derivative (PD) controller. Due to the popularity of PID controllers in industrial applications, most of the development of fuzzy controllers in the past decade has revolved around fuzzy PID controllers [13]. PI- and PD-type fuzzy controllers have gained prominence for almost two decades now because of their simple structure, ease of implementation, and inherent robustness [22]. Another possible implementation of a PID-type FC requires the development of a popular PD-type fuzzy controller in parallel with a conventional integral (I) controller. Presently, research interest in this field is focused on the development of adaptation policies that can adapt these component fuzzy controllers in a suitable and simple fashion and can further achieve a reasonably accurate and satisfactory performance. An excellent example of applying self-organized fuzzy systems to autotune the gains of a classic PID controller online and its successful application in controlling a MIMO robot arm can be found in [23].

A two-wheeled self-balancing robot [24] system is characterized as being multiple-variable, high-order, nonlinear, strong-coupling, and unstable, so it is considered a standard research target by many modern control theory researchers [25], and many abstract control concepts such as system stability, robustness, controllability, and system anti-interference properties can be displayed via two-wheeled self-balancing robot system experiments. Recently, a two-wheeled self-balancing robot, namely, Segway [26], has been widely recognized as a powerful personal transportation vehicle. It is constructed from very sophisticated and high-quality dedicated components, such as a brushless servomotor with neodymium magnets, a precision gearbox, nickel metal hydride batteries, silica-based wheels, a digital signal processor as the main controller, motor drivers, six gyroscopes,

and several safety accessories. In contrast to this kind of high-cost human transporter, many researchers have presented low-tech self-balancing transporters and claimed that such a vehicle can be built using off-the-shelf inexpensive components. Because of its simple structure, stable operation, high energy efficiency, and environmental adaptability, it has very broad application prospects in both military and civilian areas.

In this paper, the utility and effectiveness of soft computing approaches for a two-wheeled self-balancing robot with structured and unstructured uncertainties is presented. In this approach, precompensation of a hybrid fuzzy PD controller is proposed. The control scheme consists of a fuzzy logic-based precompensator followed by fuzzy PD control. Moreover, a fuzzy supervisory controller is used to supervise conventional proportional and derivative actions such that the conventional gains are adapted online through fuzzy reasoning.

We study the GBOT1001 two-wheeled self-balancing robot produced by Googol Technology (Shenzhen) Limited, and we establish the mathematical model of this system, use fuzzy PD control theory to control the robot, and, at the same time, achieve effective control.

2. System Description

2.1. Structural Analysis of the Robot System

We can develop a linear model with the following assumptions [27].

- (1) The robot is a rigid body and does not distort during moving.
- (2) The left- and right-hand wheels are completely analogous.
- (3) Cornering forces are considered negligible.
- (4) Friction is neglected during the analysis.
- (5) Since the time constant of electric motors is small compared to the system's time constants, the motor dynamics have been neglected in the model.

The design of the two-wheeled self-balancing robot is based on a mobile single inverted pendulum. The robot is composed of a chassis carrying a DC motor coupled to a planetary gearbox for each wheel, the DSP board used to implement the controller, the power amplifiers for the motors, the necessary sensors to measure the robot's states, and the receiver for the radiocontrol unit, as well as a vertical bar. The wheels of the vehicle are directly coupled to the output shaft of the gearboxes. A GE laser (scanner) motion controller is the core of the embedded DSP board. This is composed of a floating-point DSP from Analog Devices, an FPGA, three 10-bit D/A converters, and four 12-bit A/D converters.

2.2. Dynamics Model of the System

Figure 1 [27] shows the chassis diagram of the robot. The linear movement of the chassis is characterized by the position x and the speed \dot{x} ; it is also able to rotate around the axis (pitch), a movement described by the angle θ , and the corresponding angular velocity $\dot{\theta}$. In addition, the vehicle can rotate around its vertical axis (yaw) with the associated angle δ and angular velocity $\dot{\delta}$. m is the mass of rotating masses connected to the left and right wheels, M is the mass of the chassis, R is the radius of the wheel, L is the distance between the z -axis and the center of gravity of the chassis, D is the lateral distance between the contact patches

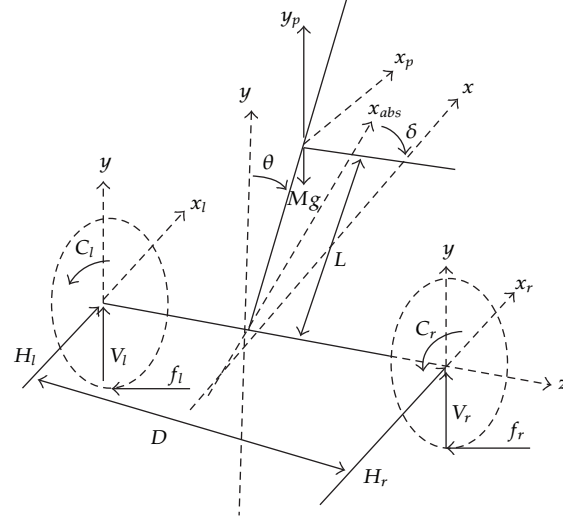


Figure 1: Chassis diagram of the robot.

of the wheels, y_p is the shift position of the wheel with respect to the y -axis, and x_p is the shift position of the chassis with respect to the x -axis.

For the left-hand wheel,

$$m \ddot{x}_l = f_l - H_l \quad (2.1)$$

$$\frac{J\omega}{R} \ddot{x}_l = C_l - f_l R.$$

For the right-hand wheel,

$$m \ddot{x}_r = f_r - H_r \quad (2.2)$$

$$\frac{J\omega}{R} \ddot{x}_r = C_r - f_r R. \quad (2.3)$$

For the chassis,

$$M \ddot{x}_p = H_l + H_r$$

$$J_p \ddot{\theta} = (V_l + V_r)L \sin \theta - (H_l + H_r)L \cos \theta - (C_l + C_r) \quad (2.4)$$

$$M \ddot{y}_p = V_l + V_r - Mg$$

$$J_\delta \ddot{\delta} = \frac{D}{2}(H_l - H_r).$$

For a premise,

$$\begin{aligned}x_p &= x + L \sin \theta, \\y_p &= L \cos \theta,\end{aligned}\quad (2.5)$$

where H_l, H_r, V_l, V_r represent reaction forces between the different free bodies [27]. Modifying the above equations and then linearizing the result around the operating point ($\theta \approx 0$, so $\sin \theta \approx \theta, \cos \theta \approx 1$), the system's state-space equations can be written in matrix form as:

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\delta} \\ \ddot{\delta} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & a_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \\ \delta \\ \dot{\delta} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ b_{21} & b_{22} \\ 0 & 0 \\ b_{41} & b_{42} \\ 0 & 0 \\ b_{61} & b_{62} \end{pmatrix} \begin{pmatrix} C_l \\ C_r \end{pmatrix}. \quad (2.6)$$

In order to derive a simpler mathematical model, we need a decoupling unit (2.7) that transforms C_l and C_r into the wheel torques C_θ and C_δ as follows:

$$\begin{pmatrix} C_l \\ C_r \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix} \begin{pmatrix} C_\theta \\ C_\delta \end{pmatrix}. \quad (2.7)$$

Some parameters of the GBOT1001 robot are as follows:

$$M = 21 \text{ kg}, \quad m = 0.42 \text{ kg}, \quad R = 0.106 \text{ m}, \quad L = 0.3 \text{ m}, \quad D = 0.44 \text{ m}, \quad g = 9.8 \text{ m/s}^2. \quad (2.8)$$

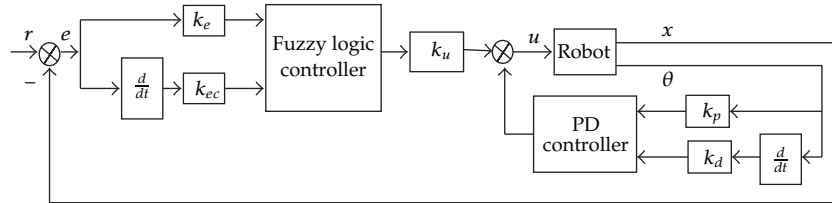
Then the state-space equations for the vehicle can be written as two different systems [28]: one system is the model (2.9) of an inverted pendulum, which describes the rotation about the z -axis, and the other is the model (2.10) of rotation, which describes the rotation about the y -axis. Thus, we obtain

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -23.7097 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 83.7742 & 0 \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ 1.8332 \\ 0 \\ -4.9798 \end{pmatrix} C_\theta, \quad (2.9)$$

$$\begin{pmatrix} \dot{\delta} \\ \ddot{\delta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta \\ \dot{\delta} \end{pmatrix} + \begin{pmatrix} 0 \\ 5.1915 \end{pmatrix} C_\delta. \quad (2.10)$$

Table 1: Fuzzy control rules.

E	Ec						
	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

**Figure 2:** Structure chart of the multiple fuzzy PD controllers.

3. Controller Design

3.1. Parameters of the Fuzzy PD Controller

First, we examine the controllability of the open-loop system. After some calculations, we know that the system is controllable and the state feedback matrices can be obtained using the LQR method. After a number of simulation experiments, we choose the matrices

$$Q = \begin{pmatrix} 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad R = 1. \quad (3.1)$$

Then, we use the MATLAB function $K = \text{lqr}(A, B, Q, R)$ to solve the optimality problem. Running $K = \text{lqr}(A, B, Q, R)$ in MATLAB, we can get the state feedback matrix:

$$K = [-10.1522, -5.6683, -26.5230, -4.1971]. \quad (3.2)$$

Thus, the proportion parameter is 26.5230, and the differential parameter is 4.1971.

3.2. Fuzzy PD Control Method

3.2.1. Fuzzy PD Controller Design

A fuzzy logic controller is a controller that utilizes fuzzy logic to determine a course of action. The computation of the control action is composed of four steps [28]: input scaling and

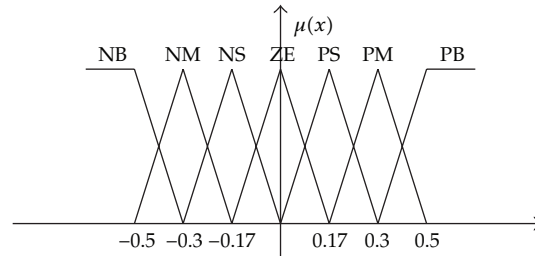


Figure 3: Membership functions of inputs.

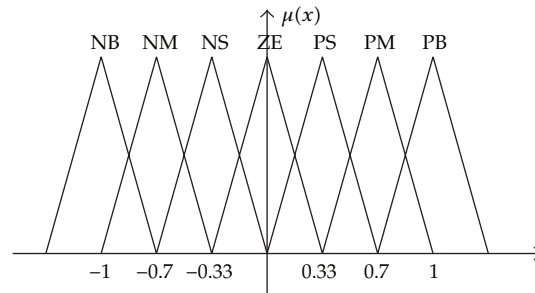


Figure 4: Membership functions of output.

shifting, fuzzification, fuzzy inference, and defuzzification. In the input scaling and shifting step, crisp inputs can be processed and scaled appropriately. In the fuzzification step, the crisp inputs are then converted into fuzzy values. After fuzzy inference, a proper control action is determined by searching through a previously established rule table. Finally, the deterministic output is defined in the defuzzification step.

In this section, a fuzzy PD controller is applied to the dynamic model of the two-wheeled self-balancing robot. Two inputs, position error $e_{(t)} = r - x$ and change of error $\dot{e}_{(t)} = \dot{r} - \dot{x}$, are fed to the fuzzy controller, while integral error is used as a conventional integral action [29]. The inputs of the fuzzy controller are the position and speed of the robot, while the angle and angle rate of the robot are controlled by the PD controller. Figure 2 shows the structure of the fuzzy PD controller.

The input ($e_{(t)}, \dot{e}_{(t)}$) and output (u) membership functions are denoted as NB, NM, NS, ZE, PS, PM, and PB. The fuzzy membership functions for inputs and output are shown in Figures 3 and 4. The control surface of the output is shown in Figure 5. Table 1 lists the fuzzy control rules.

3.2.2. Simulation Result

The control system includes a Simulink model of the feedback system [29]. The block diagram, as shown in Figure 6, consists of the subsystem 1 block with system matrices as defined in the script file. The position state is compared to a reference signal and then multiplied by its feedback gain, while the other states are simply multiplied by their respective gains, and all of these are fed back into the subsystem 1 block, which is the Simulink model of the robot, as shown in Figure 7.

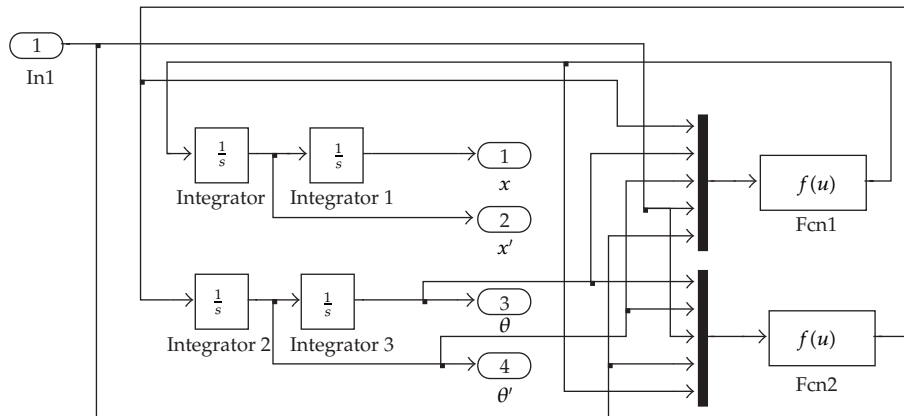


Figure 7: Simulink model of robot.

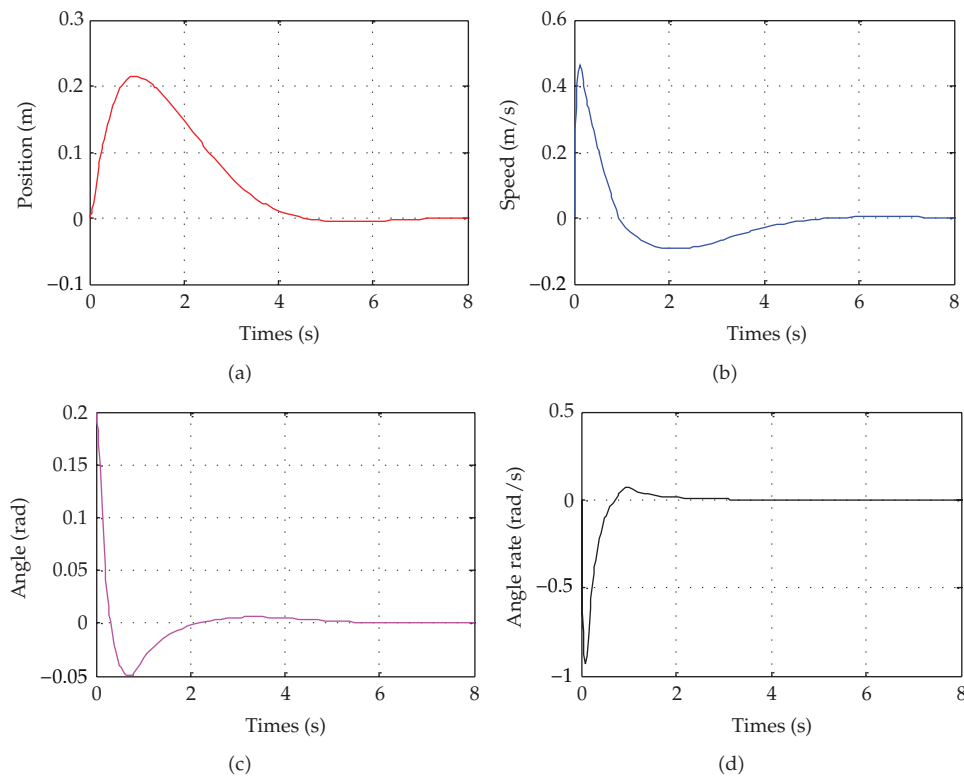


Figure 8: Graph of dip angle control.

4. Real-Time Control System

We use the fuzzy PD controller to replace the real-time control platform designed by Googol Technology Limited, so we can get the self-balancing robot's real-time control system platform, as shown in Figure 10. Compared with the simulation model, the real-time control system has a corner controller that is used to control the robot to turn around. After some

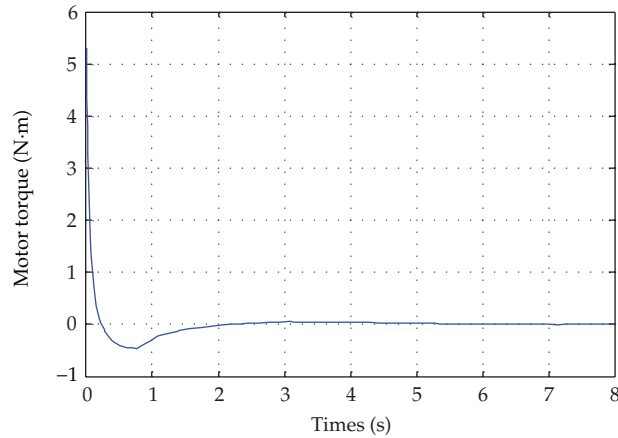


Figure 9: Graph of output motor torque.

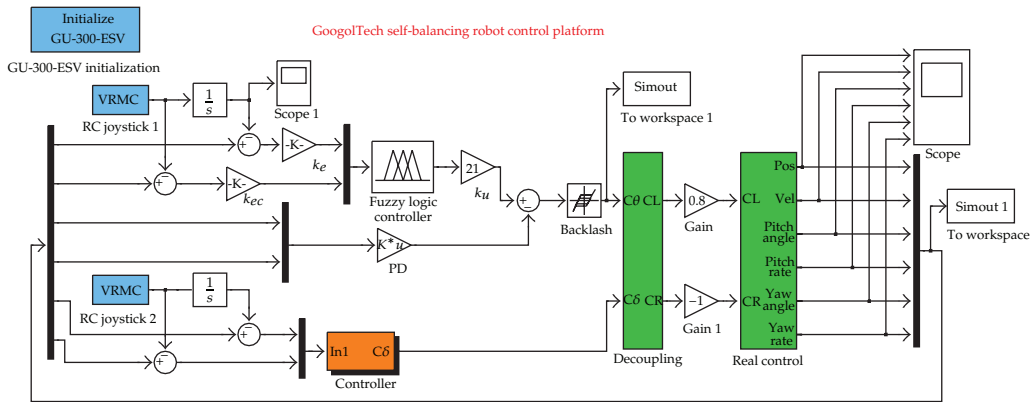


Figure 10: Actual self-balancing robot control platform.

parameter debugging, the robot is controlled effectively. Figure 11 is the robot’s real-time curve; Figure 12 is its stable control photo.

From Figure 11, we can see that the robot’s angle is very large at the beginning, about 0.31 rad, so the motor output torque is very high, that is, about 13 N · m with 1.7 m/s initial angle rate. 0.5 s later, the robot basically comes to a dynamic balance, which is 4 s faster than in the simulation. Since the ground is not flat, and there are various outside interferences, the robot does not move smoothly, and so there will be a little convulsion. After 5 s, an interference is given to the robot; from the figure we can see that the motor output torque and the angle rate increase rapidly. 0.7 s later, they all enter a stable state.

5. Conclusion

In this paper, based on the structure model of a two-wheeled self-balancing robot, a systematic mathematical model is devised according to dynamic mechanics theory. After a number of simulation experiments, we get the best Q , R , and state-feedback matrices. Based

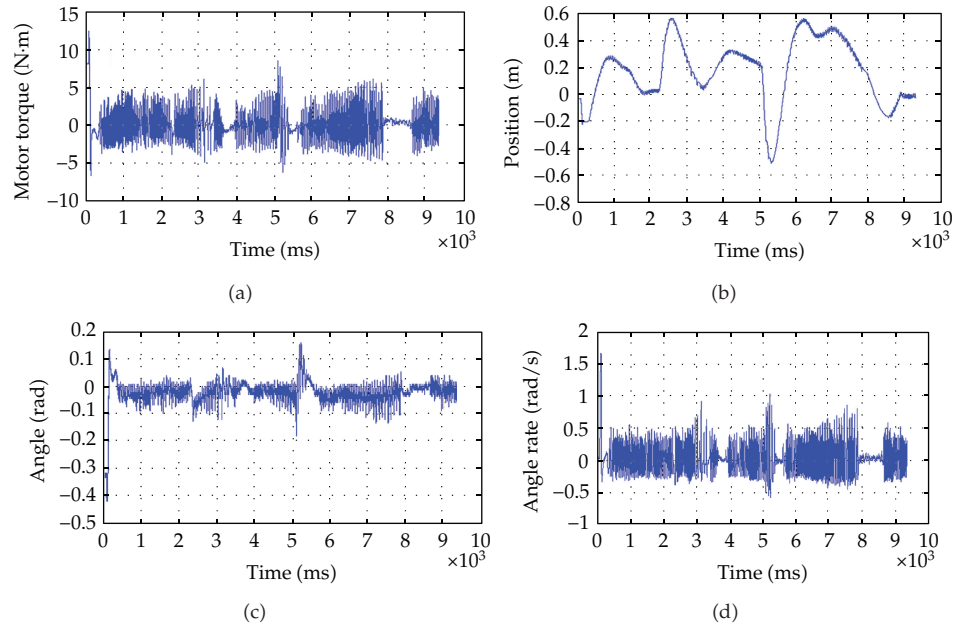


Figure 11: Real-time curve of the robot when stable.



Figure 12: Stable control photo of robot.

on traditional PD control theory and fuzzy logic control theory, a fuzzy PD controller is designed, which coordinates effectively the robust stability and speediness of the system.

Finally, a real-time control platform for the two-wheeled self-balancing robot is designed; after some parameter debugging, the robot is controlled effectively. The result indicates that the fuzzy PD control algorithm can achieve self-balanced control of the two-wheeled robot successfully and prevent the robot from falling in order to satisfy the robot's anticipated control goals and obtain a good dynamic performance.

References

- [1] K. Y. Cai and L. Zhang, "Fuzzy reasoning as a control problem," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 3, pp. 600–614, 2008.
- [2] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.
- [3] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-II," *Information Sciences*, vol. 8, no. 4, pp. 301–357, 1975.
- [4] D. Dubois and H. Prade, "Fuzzy sets in approximate reasoning. I. Inference with possibility distributions," *Fuzzy Sets and Systems*, vol. 40, no. 1, pp. 143–202, 1991.
- [5] H. Maeda, S. Asaoka, and S. Murakami, "Dynamical fuzzy reasoning and its application to system modeling," *Fuzzy Sets and Systems*, vol. 80, no. 1, pp. 101–109, 1996.
- [6] D. S. Yeung and E. C. Tsang, "Weighted fuzzy production rules," *Fuzzy Sets and Systems*, vol. 88, pp. 299–313, 1997.
- [7] J. Zhang, P. Shi, and Y. Xia, "Fuzzy delay compensation control for T-S fuzzy systems over network," *IEEE Trans on Systems, Man and Cybernetics B*, no. 99, pp. 1–10, 2012.
- [8] Q. Zhou, P. Shi, J. Lu, and S. Xu, "Adaptive output feedback fuzzy tracking control for a class of nonlinear systems," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 972–982, 2011.
- [9] G.-J. Wang, "On the logic foundation of fuzzy reasoning," *Information Sciences*, vol. 117, no. 1-2, pp. 47–88, 1999.
- [10] D. Dubois and H. Prade, "Possibilistic logic: a retrospective and prospective view," *Fuzzy Sets and Systems*, vol. 144, no. 1, pp. 3–23, 2004.
- [11] L. Wu, X. Su, P. Shi, and J. Qiu, "Model approximation for discrete-time state-delay systems in the TS fuzzy framework," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 366–378, 2011.
- [12] L. Wu, X. Su, P. Shi, and J. Qiu, "A new approach to stability analysis and stabilization of discrete-time T-S fuzzy time-varying delay systems," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 41, no. 1, pp. 273–286, 2011.
- [13] A. F. Amer, E. A. Sallam, and W. M. Elawady, "Fuzzy pre-compensated fuzzy self-tuning fuzzy PID controller of 3 DOF planar robot manipulators," in *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '10)*, pp. 599–604, Montréal, Canada, July 2010.
- [14] M. Nafar, G. B. Gharehpetian, and T. Niknam, "Using modified fuzzy particle swarm optimization algorithm for parameter estimation of surge arresters models," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 1, pp. 567–581, 2012.
- [15] W.-H. Ho, S.-H. Chen, I.-T. Chen, J.-H. Chou, and C.-C. Shu, "Design of stable and quadratic-optimal static output feedback controllers for TS-fuzzy-model-based control systems: an integrative computational approach," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 1, pp. 403–418, 2012.
- [16] K.-S. Shih, T.-H. S. Li, and S.-H. Tsai, "Observer-based adaptive fuzzy robust controller with self-adjusted membership functions for a class of uncertain MIMO nonlinear systems: a PSO-SA method," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 2, pp. 1419–1437, 2012.
- [17] J. Zhang, P. Shi, and Y. Xia, "Robust adaptive sliding-mode control for fuzzy systems with mismatched uncertainties," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 4, pp. 700–711, 2010.
- [18] C. Elmas, O. Deperlioglu, and H. H. Sayan, "Adaptive fuzzy logic controller for DC-DC converters," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1540–1548, 2009.
- [19] J. J. Craig, *Introduction to Robotics*, Addison-Wisley, Reading, Mass, USA, 2nd edition, 1989.
- [20] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, John Wiley & Sons, New York, NY, USA, 1989.
- [21] S. K. Nguang, P. Shi, and S. Ding, "Fault detection for uncertain fuzzy systems: an LMI approach," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 6, pp. 1251–1262, 2007.
- [22] T.-C. Lin, S.-W. Chang, and C.-H. Hsu, "Robust adaptive fuzzy sliding mode control for a class of uncertain discrete-time nonlinear systems," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 1, pp. 347–359, 2012.
- [23] H. B. Kazemian, "The SOF-PID controller for the control of a MIMO robot arm," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 4, pp. 523–532, 2002.
- [24] *Googol Technology Self-Balancing Robot GBOT1001 User Manual V1.0.*, Googol Technology LTD, 2007.
- [25] M. W. Spong, "Swing up control problem for the acrobot," *IEEE Control Systems Magazine*, vol. 15, no. 1, pp. 49–55, 1995.
- [26] Segway Human Transporter, 2004, <http://www.segway.com/>.

- [27] J. Wu and W. Zhang, "Research on control method of two-wheeled self-balancing robot," in *Proceedings of the 4th International Conference on Intelligent Computation Technology and Automation (ICICTA '11)*, pp. 476–479, Shenzhen, China, March 2011.
- [28] M. P. Kevin and Y. Stephen, *Fuzzy Control*, Tsinghua University Press, Beijing, China, 2001.
- [29] J. Wu and W. Zhang, "Design of fuzzy logic controller for two-wheeled self-balancing robot," in *Proceedings of the 6th International Forum on Strategic Technology (IFOST '11)*, pp. 1266–1270, Harbin, China, August 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

