International Journal of Control and Automation
Vol.9, No.2 (2016), pp.31-40
http://dx.doi.org/10.14257/ijca.2016.9.2.04

Research on Two-Wheeled Self-Balancing Robot Control Strategy Based on LQR-Fuzzy Algorithm

Junfeng Wu[1, 2], Xiaohan Wang[2] and Haiying Wang[2]

1. *Heilongjiang University of Science and Technology, Harbin, China*
2. *Harbin University of Science and Technology, Harbin, China*

E-mail: wu_jf@hrbust.edu.cn

Abstract

In this paper, we take two-wheeled self-balancing robot as the experimental object, designing an advanced controller which not only solves the defect of the over-reliance on precise mathematical model for modern control theory, but also solves the problem of complex handling issues of the fuzzy controller under multi input and multi output situation. The designing idea is: firstly, a fusion function is gained through the feedback matrix by LQR method; the fusion function is adopted to reduce the input dimensions of the fuzzy controller, thus to control the system. Secondly, due to the quantization factor and proportional factor of fuzzy control are mainly obtained by experience, we use particle swarm algorithm to optimize the quantization factor and scaling factor of fuzzy controller in order to get an ideal control effect. Thirdly, constructing two fusion functions to fuse the six dimensional input variables from the robot, which solve the existence error between the designed decoupler from theoretical calculation and the actual situation of the system, also solve the difficulties through decouper designing. Finally, proving the validity of the control strategy through the simulation and real-time control, it shares a good theoretical and practical significance.

Keywords*: self-balancing robot; fusion function; particle swarm optimization; decoupler; LQR-Fuzzy controller*

1. Introduction

Nowadays, the role of robots in the society has become more and more obvious; its product has been widely used. Due to the vulnerability of the robots and the likely influence of some uncertain factors from external environment, it belongs to a complex nonlinear system; therefore the design of its controller becomes a tough issue. Two-wheeled self-balancing robot has simple structure, and it is easy to analyze, it also shares the characteristics of nonlinearity, multivariate, instability and high coupling like other robots [1]. Hence the research of two-wheeled self-balancing robot has great significance to the field of robot controller designing; meanwhile it is also an ideal platform for testing various control algorithms [2]. The two-wheeled self-balancing robot is chosen as the experimental object in this paper, in order to solve the problem of over-reliance on precise mathematical models and the problem of complex handling issues of the fuzzy controller under multi input and multi output situation [3]. The controller is designed.

2. System Description and Modeling of the System

2.1 Mathematical Model of the System

In order to design the robot controller, the force analysis of the system is conducted and dynamic mathematical models is established, a three degrees of freedom mechanical system can be expressed by six state variables [4-5]. The horizontal direction is taken as the X axis, the vertical direction is taken as the Y axis, and the line between two wheels centers is taken as Z axis, the two center point between two wheels is the origin of the coordinate system, the coordinate system is shown in figure 1.

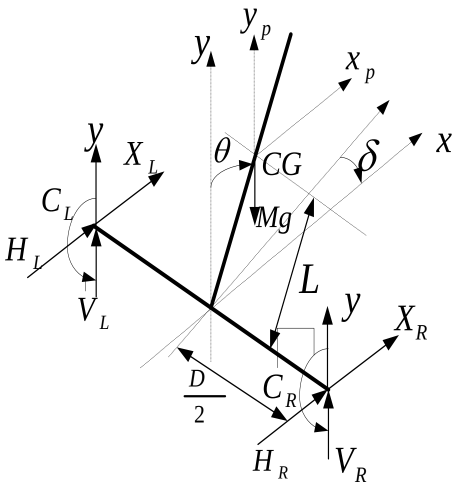


Figure 1. Self-Balancing Robot Force Analysis Diagram

2.2 The Dynamic Model Based On Newtonian Mechanics

Force analysis for the robot left wheel:

$$m\ddot{x}_l = f_l - H_l \tag{1}$$

$$\frac{J_\omega}{R}\ddot{x}_l = C_l - f_l R \tag{2}$$

Force analysis for the robot right wheel:

$$m\ddot{x}_r = f_r - H_r \tag{3}$$

$$\frac{J_\omega}{R}\ddot{x}_r = C_r - f_r R \tag{4}$$

Force analysis for the robot body:

$$M\ddot{x}_p = H_l + H_r \tag{5}$$

$$J_p\ddot{\theta} = (V_l + V_r)L\sin\theta - (H_l + H_r)L\cos\theta - (C_l + C_r) \tag{6}$$

$$M\ddot{y}_p = V_l + V_r - Mg \tag{7}$$

$$J_\delta\ddot{\delta} = \frac{D}{2}(H_l - H_r) \tag{8}$$

Among them,

$$x_p = x + L\sin\theta \tag{9}$$

$$y_p = L\cos\theta \tag{10}$$

From the above formulas a nonlinear mathematical model of the robot can be gained:

$$\ddot{x}(M + 2m + \frac{2J_\omega}{R^2}) + ML(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta) = \frac{1}{R}(C_l + C_r) \tag{11}$$

$$\begin{aligned} &J_p\ddot{\theta} = MgL\sin\theta - ML^2\ddot{\theta}\sin^2\theta - ML^2\dot{\theta}^2\sin\theta\cos\theta - \\ &(1 + \frac{L\cos\theta}{R})(C_l + C_r) + 2L(m + \frac{J_\omega}{R^2})\cos\theta\cdot\ddot{x} \end{aligned} \tag{12}$$

$$(Dm + \frac{2J_\delta}{D} + \frac{DJ_\omega}{R^2})\ddot{\delta} = \frac{1}{R}(C_l - C_r) \tag{13}$$

The self balancing linearized state equation of rob of can be gained through the system model linear processing near the equilibrium point:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\delta} \\ \ddot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -23.7097 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 83.7742 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \\ \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1.8332 & 1.8332 \\ 0 & 0 \\ -4.9798 & -4.9798 \\ 0 & 0 \\ 5.1915 & -5.1915 \end{bmatrix} \begin{bmatrix} C_l \\ C_r \end{bmatrix} \tag{14}$$

3. The Design of LQR-Fuzzy Controller

3.1 The Design of Controller

Due to the direct fusion characteristic of the output information within linear system, the fusion function is constructed, and changes the multiple state variables into comprehensive errors E and EC. The output value of fusion function after dimensionality reduction can be regarded as the input value of a two dimensional Mamdani fuzzy controller [6], both the input and the output of fuzzy controller adopt the triangle membership function, and each variable uses 7 fuzzy subsets of [NB NM NS ZE PS PM PB] for description [7-8].

The output equation of fusion function is:

$$\tilde{x} = Gx = \begin{bmatrix} \frac{k_r}{\|K\|} & 0 & \frac{k_\theta}{\|K\|} & 0 \\ 0 & \frac{k_{\dot{r}}}{\|K\|} & 0 & \frac{k_{\dot{\theta}}}{\|K\|} \end{bmatrix} x \tag{15}$$

The fuzzy rule of fuzzy controller is set up in accordance with reasoning synthesis method. Here, the optimum quadratic is adopted to gain the required input and output [9]. Therefore, the simulation diagram is shown in figure 2.

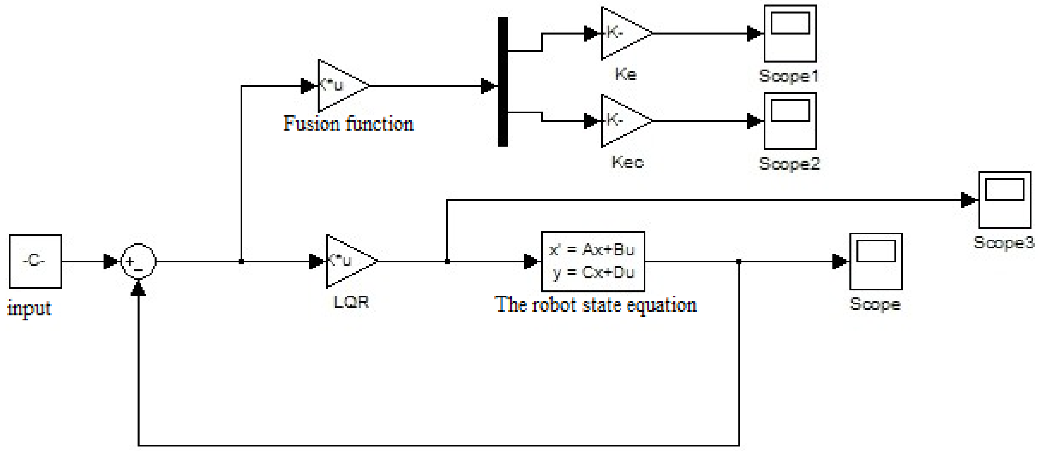


Figure 2. The Optimal the Two Times Simulation Structure Diagram

3.2 Robot Real-Time Control Test

The above LQR-fuzzy control strategy is applied into the three degrees of freedom robot system to observe and to verify the real-time control effect of algorithm. And then save the data between 10 to 20 seconds of the robot, and plot it. The effect of real-time control is shown in figure 3 as follows.

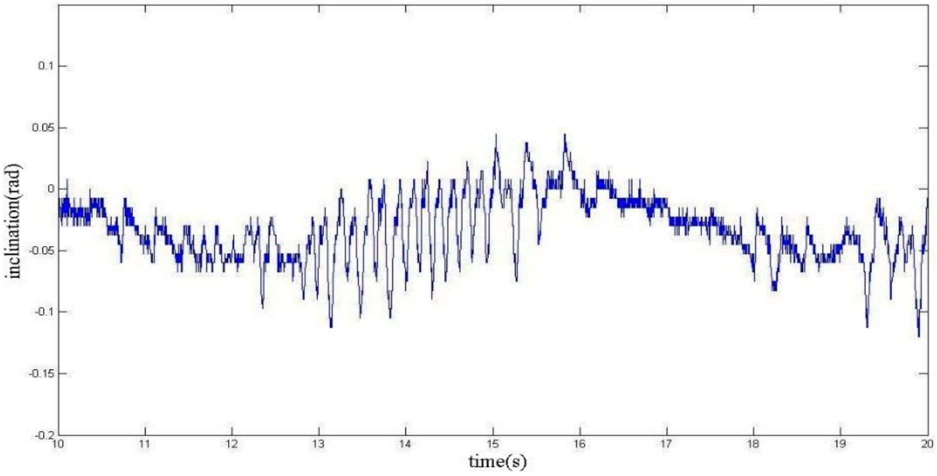


Figure 3. The Inclination Graph of LQR-Fuzzy Real-Time Control

4. Particle Swarm Optimization LQR-Fuzzy Controller

4.1 The Selection of Optimal Objective Function

This paper selects the ITAE criterion as the objective function [10-11]. ITAE criterion is integration of the absolute value between times multiplied by error. It is a more effective objective function on describing comprehensively the system dynamic performance currently. It is easy to realize the rules on the computer, and for the optimum parameter combination scheme which is gained from it can make the system with characters like good stability, less overshoot and a faster response time, etc.

4.2 The Realization of LQR-Fuzzy Control Parameter Optimization

First of all, write programs based on the idea of particle swarm optimization. Then, compile the fitness function [12]. Finally, run the program within the m file, and obtain the optimal parameter values. This is a repeated alternative simulation and optimization process. Running time will be relatively longer.

Take a set of data by experience before optimizing, taking two quantitative factors and a scale factor which are 16, 10, and 28 respectively. After the simulation, the data group the fitness value is found to be 0.562. Then, running the particle swarm program, the optimized parameters are 22.7653, 20.7528 and 24.6127 respectively, the fitness value turns into 0.421. Obviously the better fitness value can be obtained after optimization. The simulation diagrams before and after optimization are shown in figure 4 and 5.

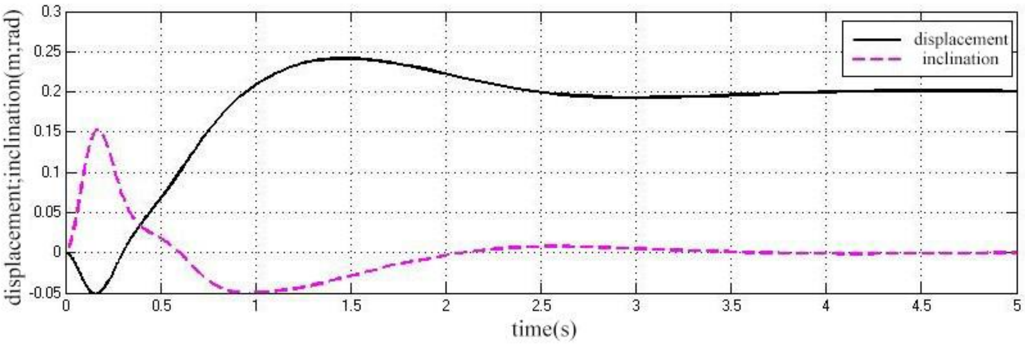


Figure 4. System Simulation Diagram Before Optimization

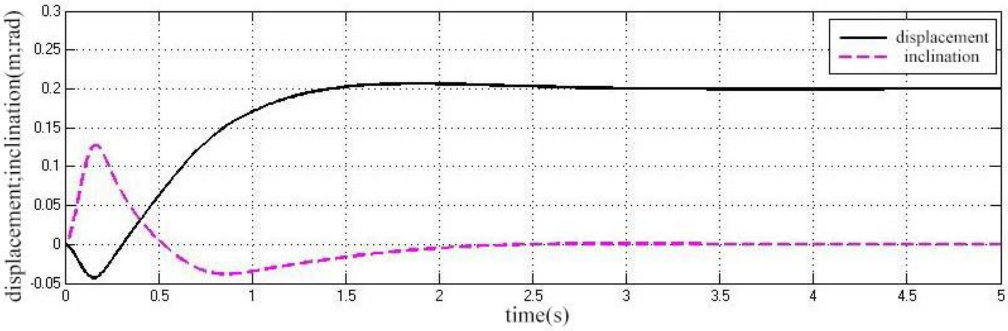


Figure 5. System Simulation Diagram after Optimization

From the above simulation diagrams can be seen that, parameters after particle swarm optimization will not only decrease the adaptive value of the system, meanwhile, we can also obviously witness the optimized system overshoot volume being significantly reduced from the simulation curve, and also speed up the convergence speed. Therefore, the particle swarm algorithm indeed plays an optimizing effect.

5. Dual Fusion of LQR-Fuzzy Controller Designing

5.1 The Construction of Fusion Function

The system state feedback matrix is firstly obtained according to the optimal control theory:

$$K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} -7.0711 & -8.0072 & -40.7360 & -5.7149 & 7.0711 & 1.1671 \\ -7.0711 & -8.0072 & -40.7360 & -5.7149 & -7.0711 & -1.1671 \end{bmatrix} \tag{16}$$

The two LQR-fuzzy controllers are designed, namely, on the basis of K_1 and K_2, constructing two fusion functions G_1 and G_2. The constructions of two fusion functions methods are basically the same [13], so only take the construction of G_1 as an example. Recording K_1 vector norm as $\|K_1\|$, $\|K_1\| = \sqrt{k_{r_1}{}^2 + k_{\dot{r}_1}{}^2 + k_{\theta_1}{}^2 + k_{\dot{\theta}_1}{}^2 + k_{\delta_1}{}^2 + k_{\dot{\delta}_1}{}^2}$.

Therefore:

$$G_1 = \begin{bmatrix} \dfrac{k_{r_1}}{\|K_1\|} & 0 & \dfrac{k_{\theta_1}}{\|K_1\|} & 0 & \dfrac{k_{\delta_1}}{\|K_1\|} & 0 \\ 0 & \dfrac{k_{\dot{r}_1}}{\|K_1\|} & 0 & \dfrac{k_{\dot{\theta}_1}}{\|K_1\|} & 0 & \dfrac{k_{\dot{\delta}_1}}{\|K_1\|} \end{bmatrix} \tag{17}$$

So the input and output of the integration function is as follows:

$$\tilde{x}_1 = \begin{bmatrix} \dfrac{k_{r_1}}{\|K_1\|} & 0 & \dfrac{k_{\theta_1}}{\|K_1\|} & 0 & \dfrac{k_{\delta_1}}{\|K_1\|} & 0 \\ 0 & \dfrac{k_{\dot{r}_1}}{\|K_1\|} & 0 & \dfrac{k_{\dot{\theta}_1}}{\|K_1\|} & 0 & \dfrac{k_{\dot{\delta}_1}}{\|K_1\|} \end{bmatrix} x \tag{18}$$

5.2 The Confirmation of Fuzzy Rules

Control strategies adopted the synthesis reasoning method with two fuzzy controllers, firstly, extract the data, and then use the fuzzy synthesis reasoning to derive the fuzzy rules on the basis of data. By using the optimal quadratic form to obtain the input and output data [14]. Then generate he control rules of fuzzy controller 1 and the controller 2, based on the ideas of fuzzy reasoning synthesis such as table 1 and table 2.

Table 1. Fuzzy Controller-Rule List

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

Table 2. Fuzzy Controller-Rule List

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

PM	**NS**	**ZO**	**ZO**	**PS**	**PS**	**PM**	**PB**
PB	**ZO**	**PS**	**PS**	**PM**	**PM**	**PM**	**PB**

5.3 Robot Real-Time Control Test

The dual fusion LQR-Fuzzy control method is applied to the practical system, and test the practical and real-time performance of dual fusion LQR-Fuzzy control algorithm. The real-time control module is shown in the figure 6.

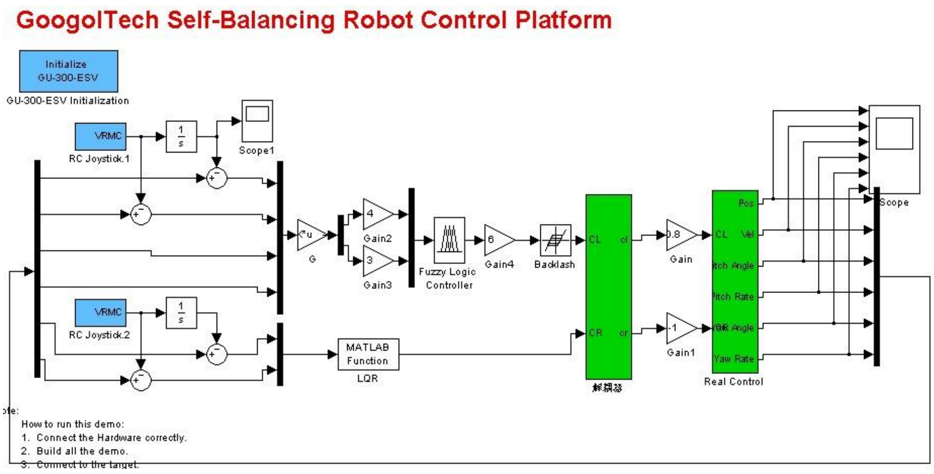


Figure 6. Simulink Model of LQR-Fuzzy Real-Time Control

The above LQR-Fuzzy control algorithms are applied to three degree freedom robot system to verify and check the practical real-time control effect of the algorithm, save the data of robots between 10 second to 20 second. The real-time effect diagram is shown in figure 7.

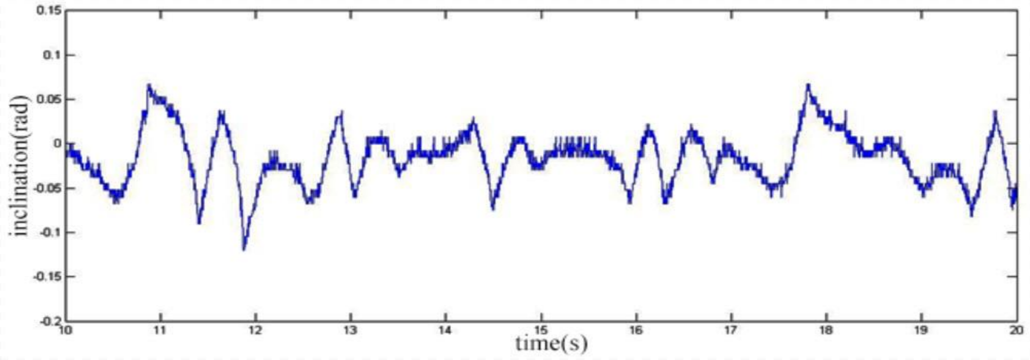
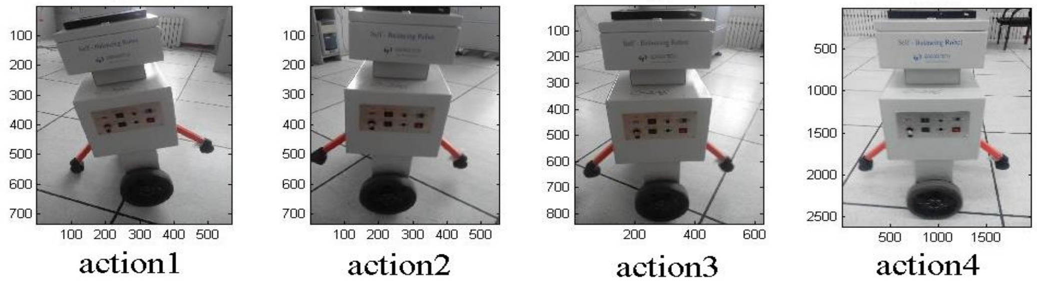


Figure 7. The Inclination Graph of Dual Fusion LQR-Fuzzy Real-Time Control

From the real time response curve of double fusion LQR-fuzzy control, it can be seen that during the process of control, the dip angle basically fluctuate within the range of [-0.05, 0.05], and the range of variation is very small. Therefore, we can say that in some cases, no decoupling to the system also can achieve the expected effect of control. The robot real time control effect is shown in figure 8.



action1 action2 action3 action4

Figure 8. Robot Real-Time Control Effect

6. Conclusion

This paper takes two-wheeled self-balancing robot as the experimental object, firstly, the Newton's mechanics equation is adopted to establish the system mathematical model and conduct the linear process. Then design a LQR-fuzzy controller, the LQR is adopted to generate the system feedback matrix, in order to construct the fusion function and then to reduce the fuzzy controller input dimension, lastly using the fuzzy controller to control the whole system. During the design of the above controller, the obtaining of fuzzy controller quantization factor and scaling factor are mainly relied on experience and test methods. In order to get a better control effect, using particle swarm optimization algorithm to optimize the fuzzy controller of quantization factor and scaling factor in order to get the ideal factor values. There is a certain difference between the decoupler that is designed based on actual system and the under actual system. The decoupler has also been a problem in control system designing. Therefore, this paper tries to construct two blending function to the overall integration, for the six dimensional input variables of the robot, using the fuzzy controller itself decoupling function instead of decoupler within the original system. Finally, the algorithm was verified through SIMULINK and proved to obtain a good simulation effect which has a wide range of application perspective.

Acknowledgements

Harbin application technology research and development projects (Outstanding subject leaders) No.2014RFXYJ051

References

[1] Googol Technology, “Self-Balancing Robot GBOT1001 User Manual”, **(2007)**, pp. 4-16.

[2] J. Wu and W. Zhang, “Design of fuzzy logic controller for two-wheeled self-balancing Robot”, The 6th International Forum on Strategic Technology, Harbin, China, **(2011)**, pp. 1266-1270.

[3] Z. Li, “Research on LQR-Fuzzy control algorithm of two-wheeled self-balancing robot”, Harbin: Harbin university of technology (master's degree thesis), vol. 32, **(2014)**.

[4] L. Xiao, “Two-wheel self-balancing robot modeling and intelligent control”, Harbin: Harbin university of technology (master's degree thesis), **(2011)**, pp. 60-61**.**

[5] J. Wu and S. Wang, “Application of fuzzy control in double inverted pendulum”, Journal of Harbin University of Science and Technology, vol. 4, no. 18, **(2013)**.

[6] W. Zhang, “Two rounds of self-balancing robot fuzzy PD control method research”, Harbin: Harbin university of technology (master's degree thesis), **(2012)**, pp. 18-24.

[7] J. Qu, W. Wu and J. Sun, “Triple inverted pendulum system fuzzy controller design and simulation [J]”, Journal of system simulation, vol. 3, no.16, (**2004)**.

[8] S. Jia, “Research and application of two-wheeled self-balancing robot control strategy based on adaptive neural-fuzzy algorithm”, Harbin: Harbin university of technology (master's degree thesis), **(2012)**, pp. 38-40.

[9] J. Wu and S. Jia, “T-S adaptive neural network fuzzy control applied in two-wheeled self-balancing robot”, The 6th International Forum on Strategic Technology, Harbin, China, 1023-1026, August, **(2011),** pp. 22-24.

[10] Z. Li, “Study on the fuzzy control of inverted pendulum system”, Xi'an: xi'an university of science and technology building (a master's degree thesis), 42-54, **(2007)**

[11] S. Chen, “Using SIMULINK and the optimization toolbox for PID parameter optimization”, Journal of modern electronic technology, vol. 9, no. 11 **(2002)**

[12] J. Wu, “Yuxin Liang. Research on the control method of two-wheeled self-balancing robot system based on variable structure control”, Journal of ICIC Express Letters, Part B: Applications, vol. 3, no. 5, **(2012)**, pp. 1185-1192.

[13] X. Xu, C. Liu, Z. Song and X. Song, “Fuzzy decoupling controller design and application”, Journal of hebei industrial science and technology, vol. 1, no. 25, **(2008).**

[14] J. Wu, W. Zhang and S. Wang, “Two-wheeled self-balancing robot with the fuzzy PD control method”, Mathematical Problems in Engineering, **(2012),** pp. 112-125.