BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **15**, No 2

Sofia • 2015 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2015-0035

A Controller Combining Positive Velocity Feedback with Negative Angle Feedback for a Two-Wheeled Robot

Lingyan Hu[1,2], *Henry Leung Ieee*[2], *Shaoping Xu*[1], *Hua Zhang*[1]

[1]*School of Information Engineering, Nanchang University, Nanchang City, 330031 China.*
[2]*Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB, Canada, T2N 1N4.*
Emails: lyhu@sce.carleton.ca xushaoping@ncu.edu.cn

Abstract: *The two-wheeled robot is a nonlinear system of multi-variables, higher-order and strong coupling. This paper presented a PID Controller with Double Loops (PCDL) to control the tilt angle and velocity of a two-wheeled robot. The angle controller is the regular negative feedback, while the velocity control is the positive feedback. The Double Loops work cooperatively to endow the system with strong anti-interference ability. The stability of the whole system is analyzed and the criterion of the system stability is developed. The simulation and experiments showed that the two-wheeled robot can self-balance and move at an expected velocity and the system has strong anti-interference ability.*

Keywords: *Two-wheeled robot, Double Loops, positive feedback, stability.*

1. Introduction

The two-wheeled robot is a new kind of a robot with two wheels equipped side by side. It has been fast developed in recent year thanks to its small size and flexibility. It has great application prospects in both civilian and military areas. For example, the two-wheeled police vehicles made by Segway Company are used in Security patrols [1]. What is more important, the two-wheeled robot is a complex nonlinear system with multi-variables, high order and strong-coupling characteristics [2]. It is an important model for researchers to test new control strategies and algorithms [3].

Many control policies based on modern control theory, such as state feedback [4], pole assignment [5] and LQR [6] are proposed to control the two-wheeled robot. The performance of these control methods depends on the model accuracy. It is difficult to choose the appropriate parameters and the robustness of the system is not good. PID controller is widely used in many areas in industry because an accurate model is not required. Tsai [7] proposed an adaptive neural network controller for two-wheeled scooter. Nasir, Pekan, Ahmad et al. [8] presented investigations on the performance comparison of Fuzzy Logic Controller (FLC) and PID controller for a highly nonlinear two-wheeled balancing robot. They can tune the gains of PID controller autonomously, however the computation is complicated and the real-time performance is poor.

This paper presents a PID Controller with Double Loops (PCDL) for a two-wheeled robot which combines the positive velocity feedback control and negative angle feedback control. The double loops worked cooperatively. Though the angle sensor had zero deviation, the robot could keep self-balance and move with the expected velocity. The stability of the whole system was analyzed and the criterion of stability was developed, which was never seen in previous similar researches. The simulation and the actual experiments both showed that the two-wheeled robot could be self-balanced and move with expected velocity under PCDL even when any disturbances existed. In addition, the system inherited the advantages of the traditional PID controller, such as easy implementation, good robustness and independence on the accurate model.

2. Modeling

In order to develop an efficient controller for the two-wheeled robot, the system dynamics has to be analyzed. The air friction and the friction of the wheel axis are ignored. Let us assume that there is no lip between the wheels and the ground, and the wheels always stay in contract with the ground. Fig. 1 shows the free body diagram of the robot. The two-wheeled robot has similar behavior with a pendulum on a cart, accordingly the modeling analysis is similar to an inverted pendulum [9].

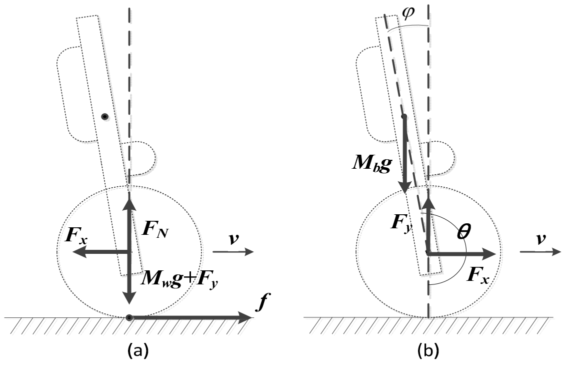


Fig. 1. A free body diagram of the two-wheeled robot: For the wheel (a); for the chassis (b)

2.1. Dynamics of the wheel

For the wheel the vertical forces applied to the wheel are shown in Fig. 1a. They are balanced and the next equation is obtained:

(1) $$F_N = M_w g + F_y,$$

where F_N is the force applied to the wheel from the ground, F_y is the vertical force applied to the wheel from the chassis, M_w is the wheel mass, g is the acceleration of gravity.

Summing the forces of the wheel in the horizontal direction, we obtain

$$M_w\dot{v} = f - F_x, \tag{2}$$

where v is the wheel velocity, f is the friction between the ground and the wheel, and F_x is the horizontal force applied to the wheel by the chassis.

The torque balance equation of wheel rotation can be expressed as

$$I_w\ddot{\theta}_w = T - fr, \tag{3}$$

where I_w is the inertia of the wheel, r is the radius of the wheel, θ_w is the angular velocity of the wheel, T is the output torque of the wheel motor.

The torque equation of the motor is as follows:

$$T = \frac{k_T U - k_T k_e \dot{\theta}_w}{R}, \tag{4}$$

where k_T is the motor torque constant, k_e is the motor back EMF constant, U is the voltage of the motor, and R is the total resistance of motor's armature circuit.

Eliminating T and f in (2)-(4) and substituting $\dot{\theta}_w = v/r$ in the equations, F_x can be expressed as

$$F_x = \frac{k_T U}{Rr} - \frac{k_T k_e v}{Rr^2} - \left(M_w + \frac{I_w}{r^2}\right)\dot{v}. \tag{5}$$

2.2. Dynamics of the chassis

For the chassis of the robot, the free body diagram is shown in Fig. 1(b). Summing the forces on the chassis in the horizontal direction, we have

$$F_x - M_b\dot{v} = -M_b l\ddot{\theta}\sin\theta - M_b l\dot{\theta}^2\cos\theta, \tag{6}$$

where M_b is the mass of the chassis, l is the distance between the centroid of the wheel and the chassis.

From Equations (5) and (6) we can eliminate F_x and get the following equation:

$$\left(M_w + \frac{I_w}{r^2} + M_b\right)\dot{v} + \frac{k_T k_e v}{Rr^2} - M_b l\ddot{\theta}\sin\theta - M_b l\dot{\theta}^2\cos\theta = \frac{k_T U}{Rr}. \tag{7}$$

Summing the forces on the chassis in a vertical direction, we have

$$F_y - M_b g = -M_b l\dot{\theta}^2\cos\theta - M_b l\ddot{\theta}\sin\theta. \tag{8}$$

According to the torque balance of the chassis, we have

$$-F_y l\sin\theta - F_x l\cos\theta = I_b\ddot{\theta}, \tag{9}$$

where I_b is the chassis inertia.

From (5), (8) and (9), eliminating F_x and F_y gives

$$(10)\qquad \left(I_b + M_b l^2\right)\ddot{\theta} + M_b g l \sin\theta = -M_b l \dot{v}\cos\theta.$$

From Fig. 1b we have $\theta = \pi + \varphi$, where φ represents a small angle from the vertical upward direction. Since the robot is being stabilized at an unstable equilibrium position where $\theta \approx \pi$, φ is always small, when the robot is self-balancing and moving. Therefore, we approximately have

$$\cos\theta = -1,\ \sin\theta = -\varphi,\ \left(\frac{d\theta}{dt}\right)^2 = 0.$$

Hence (7) and (10) can be linearized as follows:

$$(11)\ \dot{v} = \frac{M_b l}{M_w + \frac{I_w}{r^2} + M_b}\ddot{\varphi} + \frac{-k_T k_e}{Rr^2\left(M_w + \frac{I_w}{r^2} + M_b\right)}v + \frac{k_T}{Rr\left(M_w + \frac{I_w}{r^2} + M_b\right)}U,$$

$$(12)\qquad \ddot{\varphi} = \frac{M_b l}{I_b + M_b l^2}\dot{v} + \frac{M_b g l}{I_b + M_b l^2}\varphi.$$

Transforming (11) and (12) into Laplace domain we obtain:

$$sV(s) = \frac{M_b l}{M_w + \frac{I_w}{r^2} + M_b}s^2\varphi(s) + \frac{-K_T K_e}{Rr^2\left(M_w + \frac{I_w}{r^2} + M_b\right)}V(s) + \frac{K_T}{Rr\left(M_w + \frac{I_w}{r^2} + M_b\right)}U(s),$$

$$s^2\varphi(s) = \frac{M_b l}{I_b + M_b l^2}sV(s) + \frac{M_b l}{I_b + M_b l^2}\varphi(s).$$

The transfer function from the tilt angle to the input voltage for the robot is as follows:

$$(13)\ G_B(s) = \frac{\varphi(s)}{U(s)} = \frac{K_T r a M_b l s}{Rr^2\left(aI_b + aM_b l^2 - M_b^2 l^2\right)s^3 + K_T K_e\left(I_b + M_b l^2\right)s^2 - Rr^2 a M_b g l s - M_b g l K_T K_e}.$$

The transfer function from velocity to input voltage for the robot is as follows,

$$(14)\ G_S(s) = \frac{V(s)}{U(s)} = \frac{K_T r\left(I_b + M_b l^2\right)s^2 - K_T r M_b g l}{Rr^2\left(aI_b + aM_b l^2 - M_b^2 l^2\right)s^3 + K_T K_e\left(I_b + M_b l^2\right)s^2 - Rr^2 a M_b g l s - M_b g l K_T K_e},$$

where $a = M_w + \frac{I_w}{r^2} + M_b$.

3. Double-loop PID control algorithm

PID controller has extensive applications in the industrial field for its good robustness and easy tuning of the controller parameters. However, each PID controller can be used to control only one variable. In a two-wheeled control system, we have to control both the tilt angle and the velocity of the robot. Hence, two PID controllers are needed. Based on a double-loop controller for inverted pendulum, we designed a PID Controller with Double Loops (PCDL) for a two-wheeled robot similarly. One of the loops is for balance control (tilt angle control) and the other loop is for velocity control. The sum of the controllers output is the input voltage of the motor to drive the robot moving. The diagram for PCDL is shown in Fig. 2. $R_B(s)$ and $R_S(s)$ are the command signals of the angle and velocity of the robot respectively. $D_B(s)$ and $D_S(s)$ are the PID controllers of the angle and velocity respectively.

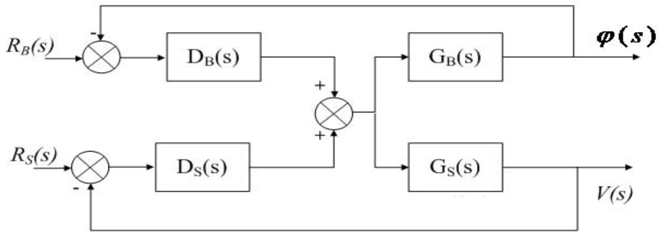
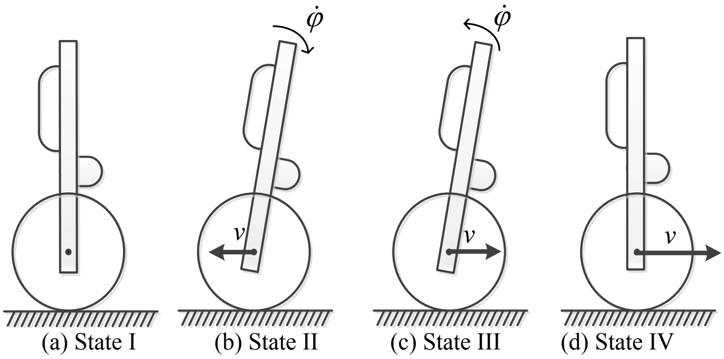


Fig. 2. The diagram of PCDL

(15) $$D_B(s) = K_{pb} + K_{db}s,$$

(16) $$D_S(s) = \frac{K_{ps}s + K_{is}}{s}.$$

The tilt angle is controlled by a PI controller whose gains (K_{pb} and K_{db}) are positive, while the velocity is controlled by a PD controller whose gains (K_{ps} and K_{is}) are negative. In this way PCDL combines the negative angle feedback and positive velocity feedback. Taking the acceleration as an example, we can use Fig. 3 to explain how PCDL works and how the robot accelerates and self-balances.



(a) State I (b) State II (c) State III (d) State IV

Fig. 3. The principle of PCDL

We assume that the robot is expected to accelerate from 0 up to 0.5 m/s. We take the rightward direction as positive for velocity, and the counter-clockwise direction as positive for angle. At the very beginning, the robot is shown as state I. The angle and velocity are both zero at this moment. When the velocity command of 0.5 m/s is sent to the controller, the error between the command and the actual velocities is positive. Accordingly the output of the velocity controller is negative which makes the wheel turn left. At the same time, because of the inertia, the robot will lean to the right, which is shown as state II. Then the tilt controller begins to work. It makes the wheel roll to the right direction, so that the robot can keep balancing which is shown as state III. At last the robot returns to balance gradually and the velocity reaches the command value (0.5 m/s), which is shown as state IV.

Normally the positive feedback would lead to system instability. However, here the positive velocity controller can be successfully used to control the two-wheeled robot move with the expected velocity. Furthermore, the two control loops work cooperatively and they endow the system with strong anti-interference ability. The system stability will be further discussed theoretically in the next section.

4. Stability analysis of the control system

In Section 2 we discussed the transfer functions from the tilt to the input voltage and from velocity to input voltage, which are $G_B(s)$ and $G_S(s)$.

Without loss of generality, we replaced the parameter variables with the actual parameters of the two-wheeled robot in this paper in order to simplify the equations. The actual parameters of the two-wheeled robot are shown in Table 1. Substituting the parameter values into (13) and (14), we have simplified $G_B(s)$ and $G_S(s)$ as follows:

$$G_B(s)=\frac{\varphi(s)}{U(s)}=\frac{104.9s}{s^3+0.04405s^2-384.3s-94.88}, \tag{17}$$

$$G_S(s)=\frac{V(s)}{U(s)}=\frac{2.368s^2-513.9}{s^3+0.04405s^2-384.3s-94.88}. \tag{18}$$

Calculating the poles of $G_B(s)$ and $G_S(s)$ gives

$$s_1=19.7;\ s_2=-19.5;\ s_3=-0.2469.$$

Because they have poles in the right complex plane, the system is unstable. We have to design the controller for the two-wheeled robot.

Table 1. The parameters of the two-wheeled robot

Various parameters	Symbol	Quantity	Unit
Motor armature resistance	R	1.875	Ω
Electromagnetic constant	K_e	0.06	V per 1 rad/s
Torque constant	K_T	0.163	(N.m)/A
Wheel radius	r	0.0325	m
The distance of the connection point to the body center of gravity	l	0.04	m
The quality of the wheels	M_w	0.05	kg
The quality of the vehicle	M_b	0.42	kg
Wheel rotation inertia	I_w	0.00002	kg.m^2
Vehicle body rotation of inertia	I_b	0.00022	kg.m^2
Acceleration of gravity	g	9.8	N/m^2

To control the two-wheeled robot self-balance and move with the expected velocity, the PCDL is designed as shown in Fig. 2. Then the close-loop transfer functions of the angle and velocity are expressed as follows:

$$G_{BC}(s)=\frac{\varphi(s)}{R_B(s)}=\frac{D_B(s)G_B(s)}{1+D_B(s)G_B(s)+D_S(s)G_S(s)}, \tag{19}$$

$$G_{SC}(s)=\frac{V(s)}{R_S(s)}=\frac{D_S(s)G_S(s)}{1+D_B(s)G_B(s)+D_S(s)G_S(s)}. \tag{20}$$

Substituting (15), (16), (17), (18) into Equations (19) and (20), we have

$$G_{BC}(s)=\left(104.9K_{db}S^3+104.9K_{pb}S^2\right)/\left[S^4+\left(104.9K_{db}+2.368K_{ps}+0.04405\right)S^3+\right.$$

$$\left.+\left(104.9K_{pb}+2.368K_{is}-384.3\right)S^2-\left(513.9K_{ps}+94.88\right)S-513.9K_{is}\right],$$

$$G_{SC}(s)=\left(2.368K_{ps}S^3+2.368K_{is}S^2-513.9K_{ps}S-513.9K_{is}\right)/\left[S^4+\left(104.9K_{db}+2.368K_{ps}+0.04405\right)S^3+\right.$$

$$\left.+\left(104.9K_{pb}+2.368K_{is}-384.3\right)S^2-\left(513.9K_{ps}+94.88\right)S-513.9K_{is}\right].$$

$G_{BC}(s)$ and $G_{SC}(s)$ have the same denominator, which means that the two close-loop transfer functions have the same characteristic equation. Therefore, if a group of PID parameters meet the stability requirements of one close-loop transfer function, the parameters also meet the stability requirements of another.

According to Routh's stability criterion [10], the system is stable only if the parameters of the PCDL satisfy the following stable conditions.

$$K_{db}>-0.022K_{ps}-0.0004, \tag{21}$$

$$K_{pb}>-0.02K_{is}+3.66, \tag{22}$$

$$K_{ps}<-0.185, \tag{23}$$

$$K_{is}<-0, \tag{24}$$

$$a_1a_2-a_0a_3>0, \tag{25}$$

$$a_3(a_1a_2-a_0a_3)-a_1^2a_4>0, \tag{26}$$

where,

$$a_0=1,$$

$$a_1=104.9K_{db}+2.368K_{ps}+0.04405,$$

$$a_2=104.9K_{pb}+2.368K_{is}-384.3,$$

$$a_3=-513.9K_{ps}-94.88,$$

$$a_4=-513.9K_{is}.$$

Although these conditions are only for the two-wheeled robot whose parameters are shown in Table 1, the method for developing the criterion of system stability is general.

5. Simulation

In this section we did several simulation experiments to verify the effect of PCDL designed in this paper with Simulink.

5.1. Simulation for balance and velocity control

The Simulink block diagram for the two-wheeled robot with PCDL is shown in Fig. 4. In this simulation the robot was expected to self-balance and move with the specified velocity which is 0.5 m/s. The "Balance input signal" block is the command angle signal, and it is set to zero. "Velocity input signal" block is the command velocity, and it is a step signal whose amplitude is 0.5. The "Balance controller" block and the "Velocity controller" block are the angle and velocity controllers of PCDL designed in Section 2. The "Balance object" block and the "Velocity object" block are the angle and velocity transfer functions of the robot. According to the stability analysis in Section 4, the two-wheeled robot system with PCDL can be stable only if the parameters satisfy the stable conditions (21)-(26). We determined a set of parameters according to the conditions as following: K_{pb}=45, K_{db} = 3, K_{ps} = –12, K_{is} = –10. The initial angle and velocity were both zero.

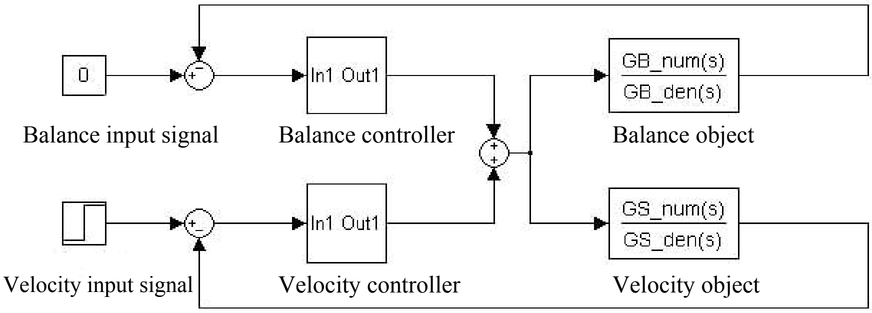
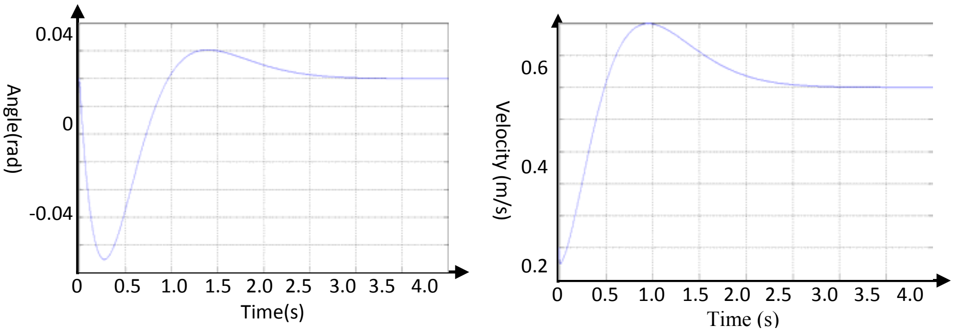


Fig. 4. Simulink block diagram for a two-wheeled robot system



(a)

(b)

Fig. 5. Simulation results for a two-wheeled robot system: Angle of the robot (a); velocity of the robot (b)

The angle and the velocity output in the simulation are shown in Fig. 5. According to Fig. 5a, the robot stood initially upright, and leaned to the right after the robot began to move. The robot recovered the balance because of the negative angle feedback of the angle controller. As seen in Fig. 5b, although the expected velocity is 0.5 m/s, the robot moved towards the negative direction at first and then moved towards the expected direction. At last, the velocity reached the expected value under PCDL. The simulation results are matched with the theoretical analysis in Section 3.

5.2. Simulation of the condition when the angle sensor has a zero deviation

In actual applications the angle sensor has always zero deviation which is shown in Fig. 6. It means that the angle measurement deviates from the actual angle, which will cause that the actual angle could not reach the command signal with a normal negative feedback controller. The PCDL has two control loops which work cooperatively. It can keep the robot self-balancing and moving with the expected velocity even if the angle sensor has zero deviation. This cannot be realized only with an angle negative feedback controller.

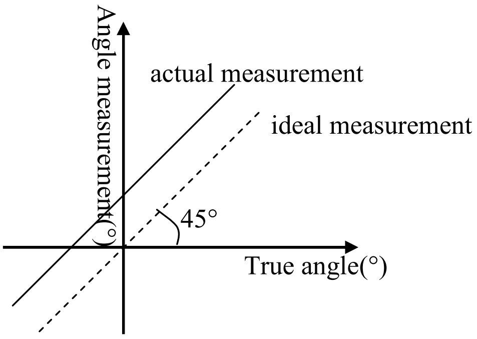


Fig. 6. The tilt angle measurement with zero deviation

The Simulink block diagram for the system whose angle sensor has zero deviation is shown in Fig. 7. The parameters for the system are the same as in the above simulation, except that the angle feedback to the controller has a deviation of 0.02 rad. The simulation result is shown in Fig. 8. The robot kept self-balancing and moving with the expected velocity of 0.5 m/s. The angle can converge to 0 even when the angle feedback to the controller has deviation of 0.02 rad. Hence PCDL has the advantage of strong anti-disturbance from the angle sensor, which is very important for the actual application.

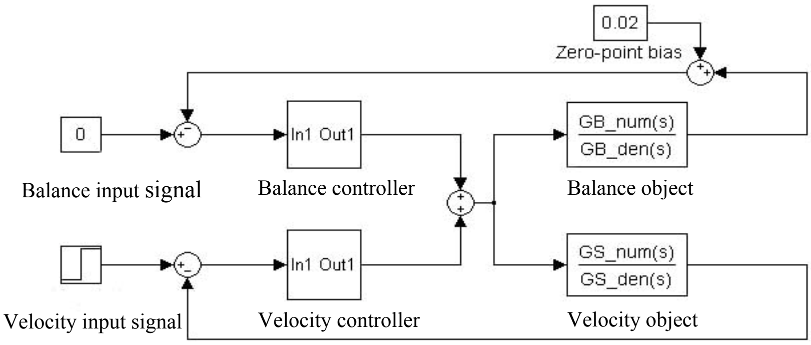
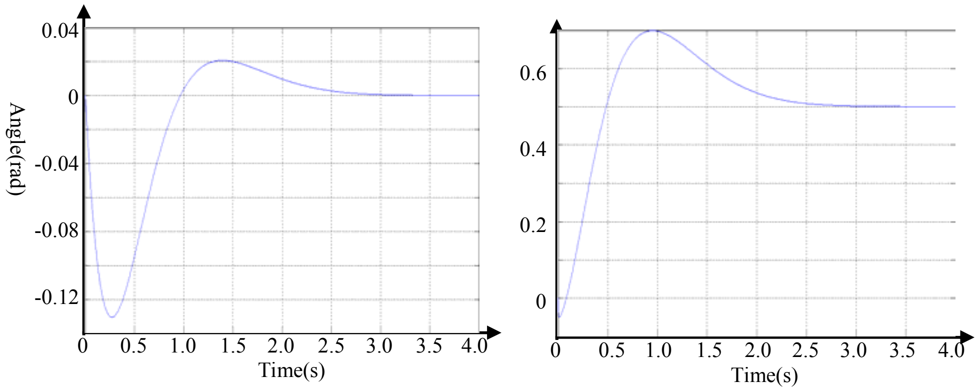


Fig. 7. Simulink block diagram for the system with zero deviation



(a) (b)

Fig. 8. Simulation result for a two-wheeled robot system with zero deviation: Angle of the robot (a); velocity of the robot (b)

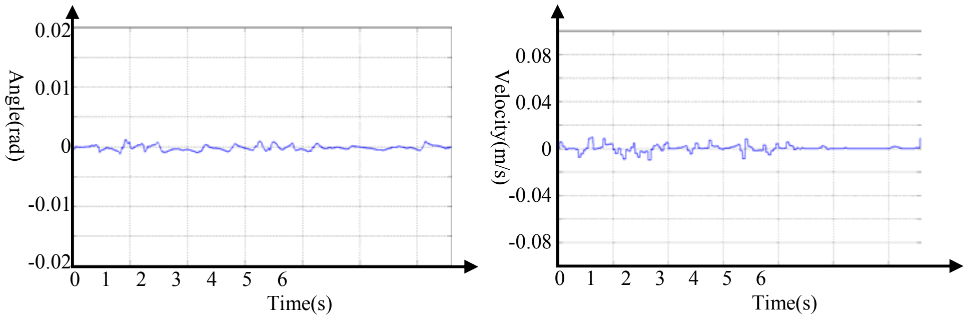
6. Experiment

We designed a two-wheeled robot described in [11] and it was used to do the experiments in this paper. The PCDL algorithm was coded and downloaded in the CPU of the robot. The angle was measured by fusing the gyro signal with an accelerometer signal with the help of Kalman filtering method. The velocity was measured with a photoelectric encoder. According to the parameters used in the simulation, we did three experiments.

Firstly, the initial angle and velocity were both zero and the command velocity was set zero, which means the robot was expected to stand still and keep balance. The experimental result is shown in Fig. 9. The actual angle and velocity of the robot were approximately zero, which meant the robot self-balanced and stood still as expected.

In the second experiment the robot stood still and kept balance initially, then we gave the robot a disturbance by pushing the robot. The robot was tilted at first and then recovered to balance soon. The experimental result is shown in Fig. 10. To compare conveniently, we drew the angle and velocity outputs in the figure. The robot stood still and kept balance at first. At 2.7-th second, the robot was pushed and tilted about 0.045 rad. After about one second, the robot recovered balance and the angle was converged to zero.

In the third experiment, the initial angle and velocity were both zero, and the command velocity was set to 0.5 m/s. The robot moved with 0.5 m/s velocity and kept balance under PCDL. The experimental result is shown in Fig. 11. Although the angle measurement has zero deviation in the actual experiment, the robot can still keep balance. The experimental results are matched with the simulation ones.



(a)

(b)

Fig. 9. Experimental results (the command velocity is set to 0): Angle of the robot (a); velocity of the robot (b)

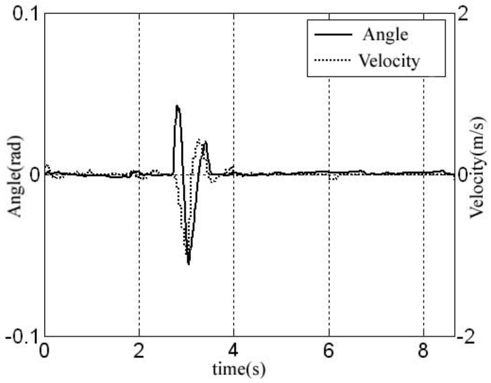
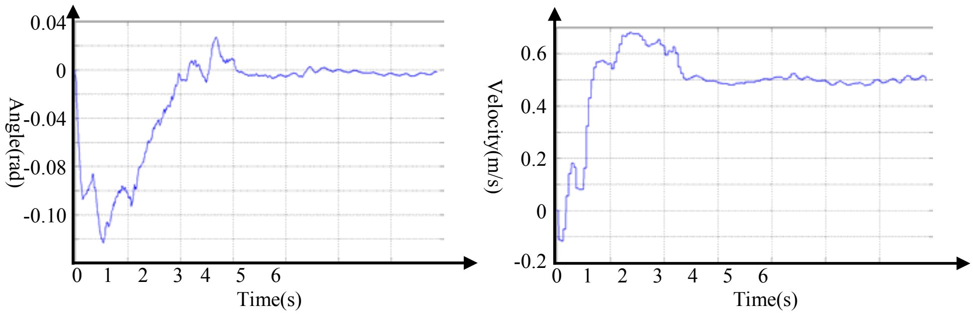


Fig. 10. Experimental result with disturbance



(a)

(b)

Fig. 11. Experimental results (the command velocity is set to 0.5m/s)): Angle of the robot (a); velocity of the robot (b)

7. Conclusion

The paper analyzes the model of a two-wheeled robot and then presents PCDL for it to ensure the robot self-balance and movement at expected velocity. We have analyzed the system stability and gave the stability criterion. According to the stability criterion we can easily determine the parameters of PCDL. The simulation and experiment both showed that the robot can self-balance and move with an expected velocity under PCDL. Furthermore, the PCDL has strong anti-interference ability because the two control loops work cooperatively. The simulation showed that the robot can self-balance and move with expected velocity even if the angle sensor has zero deviation and the experiment also showed that the robot can keep self-balancing with disturbance under PCDL. In addition, the system inherits the advantages of the traditional PID controller, such as easy implementation, good robustness and independence on the accurate model.

***Acknowledgements*:** This work was partially supported by the National Natural Science Foundation of China under Grants 51165033, 61261010 and 61163023, by the Science and Technology Department of Jiangxi Province of China under Grants 20133BBE50029, the Young Scientists Training Program of Jiangxi Province (20142BCB23001) the International Scientific Cooperation Projects of Jiangxi Province (20141BDH80001)

References

1. Nguyen, H. G., J. Morrell, K, Mullens et al. Segway Robotic Mobility Platform. – In: Proc. of SPIE (International Society for Optical Engineering), Philadelphia, PA, United States, December 2004, pp. 207-220.
2. Oryschuk, P., A. Salerno, A. M. Al-Husseini, J. Angeles. Experimental Validation of an Underactuated Two-Wheeled Mobile Robot. – IEEE/ASME Transactions on Mechatronics, Vol. **14**, 2009, No 3, pp. 252-257.
3. Salerno, A., J. Angeles. A New Family of Two-Wheeled Mobile Robots: Modeling and Controllability. – IEEE Transaction on Robotics, Vol. **23**, 2007, No 1, pp. 169-173.
4. Deng, M., A. Inoue, K. Sekiguchi, L. Jiang. Two-Wheeled Mobile Robot Motion Control in Dynamic Environments. – Robotics and Computer-Integrated Manufacturing, Vol. **26**, 2010, No 3, pp. 268-272.
5. Li, C. Q., X. S. Gao, K. J. Li. Modeling Based on Integrity Constraints and Balance Control for Two-Wheeled Robot. – Control Engineering of China, Vol. **19**, 2012, No 3, pp. 478-481.
6. Xu, C., M. Li. The System Design and LQR Control of a Two-Wheels Self-Balancing Mobile Robot. – In: International Conference on Electrical and Control Engineering, Yichang, China, September 2011, pp. 2786-2789.
7. Tsai, C. C., H. C. Huang, S. C. Lin. Adaptive Neural Network Control of a Self-Balancing Two-Wheeled Scooter. – IEEE Transactions on Industrial Electronics, Vol. **57**, 2010, No 4, pp. 1420-1428.
8. Nasir, A. N. K., M. Pekan, M. A. Ahmad et al. Performance Comparison between Fuzzy Logic Controller (FLC) and PID Controller for a Highly Nonlinear Two-Wheels Balancing Robot. – In: First International Conference on Informatics and Computational Intelligence, Bandung, Indonesia, December 2011, pp. 176-181.
9. Ooi, R. C. Balancing a Two-Wheeled Autonomous Robot. Final Year Thesis, The University of Western Australia, School of Mechanical Engineering, 2003.
10. Saeed, S. H. Automatic Control Systems. Delhi, Katson Publishers. 2008. ISBN 978-81-906919-2-5.
11. Hu, L.Y., P. Wan, X. P. Liu et al. The Balance and Speed Control System for Two-Wheeled Robot. – Electric Drive, Vol. **43**, 2013, No 12, pp. 52-55.