Indonesian Journal of Engineering and Science, Vol. 4, No. 1, 2023
https://doi.org/10.51630/ijes.v4i1.81

A METHOD OF PD CONTROL FOR BALANCING A UNICYCLE ROBOT

HAI VAN-DONG NGUYEN[1*], THANH-DONG PHAM[2], MANH-TUAN TRAN[2], NGOC-BAO BUI[2], VAN-NGHIA PHAM[2], TRAN-NGUYEN-VU PHAM[2], VAN-THANG PHAM[2], TAN-LOC NGUYEN[2]

[1]*Faculty of Electrical and Electronics Engineering, Ho Chi Minh city University of Technology (HCMUTE), Ho Chi Minh city, Vietnam*
[2]*Faculty of High Quality Training, Ho Chi Minh city University of Technology (HCMUTE), Ho Chi Minh city, Vietnam*

**Corresponding author: hainvd@hcmute.edu.vn*

(Received: 20 December 2022; Accepted: 1 February 2023; Published on-line: 1 March 2023)

ABSTRACT: Unicycle is a robot that imitates a performance of a circus artist on a one-wheeled self-balancing bicycle. This research assumes that this model is equivalent to two separated popular models: a two-wheeled self-balancing robot and reaction wheeled inverted pendulum. On each model, we build a PD controller. Thence, we present a structure of PD controllers to balance this model at the equilibrium point. We also build an experimental unicycle robot for the laboratory. Our method is proven to work well based on both simulation and experiment.

KEY WORDS: Unicycle; PD control; MIMO under-actuated; Balancing control; Self-balance.

1. INTRODUCTION

Current research trend, balance a unicycle model (unicycle robot – UR) developing on both simulation and experiment [1]. They build a controller for the overhead disk to balance the roll angle for UR and focus on controlling the forward wheel movement. The above group has approximated the UR from under-actuated MIMO to SIMO form and applied a linearized feedback algorithm to control the UR to move forward while stabilizing the system balance. However, a complete control structure for the UR's roll and pitch angle has not been fully demonstrated.

Based on the study [2], the LQR algorithm was successfully designed and tested on both simulation and experiment. In that study, the system was successfully balanced, and the vehicle could move backward and forward. That study successfully shows the balance of the system. UR in that study can move backward and forward. Some authors have also proposed a mathematical model of UR on an inclined plane [3]. The LQR method is proven again with the ability to control the UR in the inclined plane, in which only a special case [2]. Therefore, the LQR algorithm is simple and easy to be designed. However, it requires exact parameters and a mathematical model of the system. In most cases, the real model is a grey-box or black-box model. Thence, the ability of the LQR method is not common in real experiments.

A UR model on the simulation is also balanced between PD-fuzzy and LQR conditions [4]. The above study proves that a conditional condition designed through expert experience (PD-fuzzy) can still achieve better control results than a conditional condition designed through mathematics (LQR) if selected through a suitable number. However, the UR model in the above article is a UR with wheels with a width large enough to self-correct the vehicle's tilt corresponding to the roll angle. As such, it is essentially no different from a self-balancing two-wheeler, not a true UR.

A complete study of simulation is introduced mathematical equation was introduced with three motors controlling the bicycle's roll, pitch, and yaw angles [5]. The system is stabilized in situ with linear algorithms such as LQR and PID on simulation. However, this study is only at the simulation level, not experimentally verified. Between PID and LQR, PID is utilized more in the industry [6]. However, the structure of PID is commonly used for SISO structures, while LQR can be used in SIMO or MIMO under-actuated structures. The disadvantage of the LQR method is that information on dynamic equations and system parameters must be known. In contrast, the PID method can be calibrated for a black-box model.

To apply PID control for UR, in this paper, we propose to assume UR as two popular models: a two-wheeled self-balancing robot [7] and a reaction-wheeled inverted pendulum [8]. These models are SIMO systems and have been controlled well in both simulation and experiment by PID algorithm [9, 10]. Thence, we can apply two PID control structures for UR. Besides simulation, an experimental model is created to test this PID structure for UR. This research confirms the ability to use PD control for a MIMO under-actuated model, such as UR. Also, we examine to give a survey in calibration PD parameters.

2. MATHEMATICAL MODEL OF UR

The operation of UR in 3D space is described through the angle of inclination (roll), angle of incidence (pitch), and angle of rotation (yaw), as in Fig. 1.

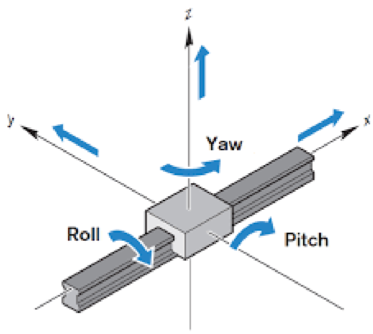


Fig. 1. Shows tilt (roll), incidence (pitch), rotation (yaw) angles

The UR model structure is shown in Fig. 2. To keep the system from falling sideways (as shown in Fig. 2a), a flywheel construction is implemented to approximate the system to a reaction-wheeled inverted pendulum form. Similarly (as shown in Fig. 4a), to keep the UR from falling forward or backward (as shown in Fig. 2b), a motor is directly attached to the wheel to approximate the system as a self-balancing robot (as shown in Fig. 3b).

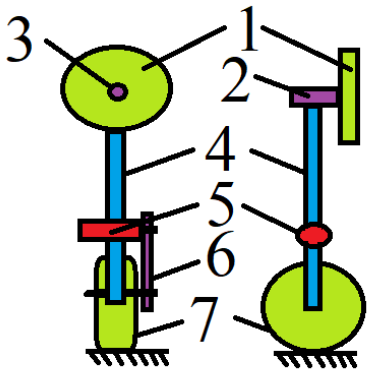


Fig. 2. Structural model UR

Explanation of the parameters in Fig. 2:

1- Reaction wheel 1 is used to stabilize the tilt angle.
2- Motor DC 1
3- Axis of DC motor 1
4- Body of UR
5- DC motor 2 to control the rotation of the wheel. From there, the angle of incidence is controlled by inertia.
6- Transmission mechanism (usually a belt) between the DC 2 motor shaft and axis of wheel.

The mathematical structure of UR is shown in Fig. 3, and the system parameters of UR are listed in Table 1.

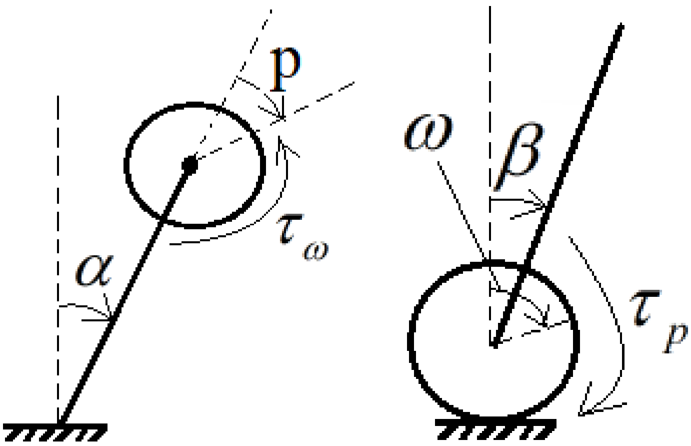


Fig. 3. Structural model UR

According to [5], when the rotational angle is small, this angle and the motor mechanism to rotate UR are not related to other state variables. In addition, in that document, because dynamic system equations are too complicated, it is impossible to present them all. The formulas are described only through Lagrange's basic formulas. The angle of incidence and inclination state variables are also assumed to be very small. Therefore, dynamic equations of the system when operating near the equilibrium point.

$$\alpha = \dot{\alpha} = p = \dot{p} = \omega = \dot{\omega} = \beta = \dot{\beta} \quad (1)$$

Table 1: Simulation model parameters *UR*

Variables/ Parameters	Description (unit)	Value
α	Tilt angle of UR (rad)	
p	Rotation angle of flywheel (rad)	
ω	Wheel rotation angle (rad)	
β	The angle of incidence of UR (rad)	
τ_ω	Torque generated by DC motor 1, acting on flywheel (Nm)	
τ_p	Torque generated by DC 2 motor, acting on the wheel (Nm)	
$\mathbf{m_1}$	Wheel weight (kg)	1
$\mathbf{m_2}$	Body mass UR (kg)	3.7
$\mathbf{m_3}$	Flywheel mass (kg)	2.66
$\mathbf{r_1}$	Wheel radius (m)	0.12
$\mathbf{r_2}$	The assumed radius of the vehicle body (m)	0.04
$\mathbf{r_3}$	Flywheel radius (m)	0.15
$\mathbf{l_2}$	Body length UR (m)	0.2
$\mathbf{l_3}$	The length of the balance bar attached to the flywheel (if any). In this paper, the group does not use (m)	0
$\mathbf{h_2}$	Distance between wheel gravitational center of the body of UR and axis of wheel (m)	0.22
$\mathbf{h_3}$	Distance between gravitational center of flywheel center and body of UR (m)	0.32
$\mathbf{s_3}$	Distance between the top of UR and the center of the flywheel	0.15

It would be abbreviated as follows:

$$M\ddot{x}+G=N\tau \tag{2}$$

In it, we have: $M(x)=\begin{bmatrix} M_{11} & M_{12} & 0 & 0 \\ M_{21} & M_{22} & 0 & 0 \\ 0 & 0 & M_{33} & M_{34} \\ 0 & 0 & M_{43} & M_{44} \end{bmatrix}$; $G(x,\dot{x})=\begin{bmatrix} 0 \\ G_2 \\ G_3 \\ 0 \end{bmatrix}$; $N=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$;

$M_{11}=J_{12}+r_1^2(m_1+m_2+m_3)$; $M_{12}=M_{21}=r_1(m_2h_2+m_3h_3+m_3s_3)$;

$M_{22}=J_{22}+J_{32}+2m_3h_3s_3+m_2h_2^2+m_3h_3^2+m_3s_3^2$;

$M_{33}=J_{11}+J_{21}+J_{31}+2m_3(h_3s_3+r_1s_3+r_1h_3)+m_3(h_3^2+s_3^2)+r_1(m_2+m_3)$;

$M_{34}=M_{43}=-J_{31}-m_3(r_3s_3+s_3^2+h_3s_3)$; $M_{44}=J_{31}+m_3s_3^2$;

$M_{55} = J_{13} + J_{23} + J_{33} + 2m_2 r_1 h_2 + m_2\left(h_2^2 + r_1^2\right)$; $G_2 = -g\beta\left(m_2 h_2 + m_3 h_3 + m_3 s_3\right)$;

$G_3 = -g\alpha\left(m_1 r_1 + m_2\left(h_2 + r_1\right) + m_3\left(h_3 + s_3 + r_1\right)\right)$; $\tau = \begin{bmatrix} 0 & \tau_\omega & 0 & \tau_p \end{bmatrix}^T$

System (2) is rewritten as follows to describe the dynamic equations of UR on simulation:

$$\ddot{x} = M^{-1}\left[N\tau - G\right] \tag{3}$$

3. PD CONTROL

PID control is commonly used in academia and industry. In the experiment, usually, the Ki component is ignored because this component is the accumulation of errors during operation. In the experiment, when the system starts to work, the initial instability of the system makes this "error accumulation" big over time. On the other hand, the PD structure (Fig. 4) is the same as the LQR structure. And, LQRstructure can control an under-actuated MIMO system if the 'controllability' of the system is proven. Therefore, the PD controller is suitable enough to control the system instead of PID or LQR structures.

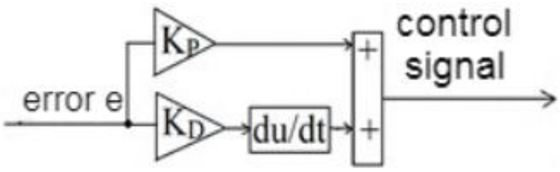


Fig. 4. Structure of a PD Controller

In this study, we consider that the UR consists of two separate SIMO systems, including:

- The first system is the reaction wheeled inverted pendulum system in which the tilt angle is controlled by a flywheel (roll). The voltage is applied to the DC motor 1, creating a torque to stably control the two state variables of flywheel rotation angle ω and tilt angle α of UR (Fig. 3a).
- The second system is the rotating wheel to control the angle of incidence (pitch). The voltage is applied to the DC motor 2, creating a torque to stably control the two state variables of wheel rotation angle and angle of incidence β of UR. This system is equivalent to the two-wheeled self-balancing robot (Fig. 3b).

Considering system 1 in Fig. 3a, if α increases, τ_ω should be decreased. Similarly, if p increases, τ_ω should be increased. Thus, the structure of the PD controller for system one is built as in Fig. 5a, with Kp1, Kd1, Kp2, and Kd2 being positive parameters. Considering system 2 in Fig. 3b, if β increases, it τ_p should be decreased. Similarly, if ω increases, τ_p should be decreased. Thus, the PD structure for system two is built as in Fig. 5b, with Kp3, Kd3, Kp4, and Kd4 as positive parameters.

For the SISO object, tuning PD control parameters is relatively straightforward. However, when using combined PD blocks, as in Fig. 5a or Fig. 5b, the calibration for each PD block also affects the other PD blocks. Therefore, parameter adjustment is more difficult than calibration SISO block.

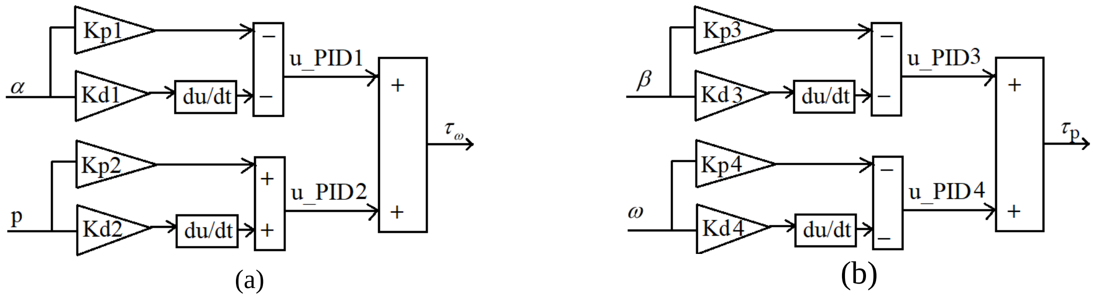


Fig.5. Structure of PD controller: (a) For system 1- (b) For system 2

4. SIMULATION

4.1. Simulation Condition

We select the initial state variable values close to the static working point in (1) as follows:

$$\alpha = 0.05; \dot{\alpha} = p = \dot{p} = \omega = \dot{\omega} = \beta = 0 \tag{4}$$

For the UR system described in (3) and the model parameters listed in Table 1, we survey the following cases (TH):

Case 1: This is the standard case. Finding a set of PD controllers is difficult. Then, we use a a genetic algorithm to find a set that can balance UR well. Based on this basic set, experiments in cases 2, and 3 will be operated by trial–and–error tests to examine the calibration of the PD controller. By using a genetic algorithm, standard parameters in case 1:

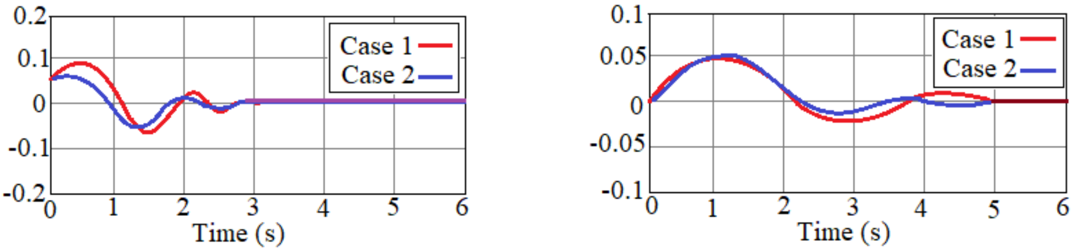
$$\begin{aligned} &Kp1 = 23.38; Kd1 = 0.24; Kp2 = 29.57; Kd2 = 2.03; \\ &Kp3 = 30.45; Kd3 = 2.89; Kp4 = 28.55; Kd4 = 0.99 \end{aligned} \tag{5}$$

Case 2: The parameters were the same as in (5), except that Kp1 was reduced to 15, compared to 19.98.

Case 3: The parameters are the same as in (5), except that Kp3 is reduced to 20, compared to 24.79.

4.2. Simulation Results

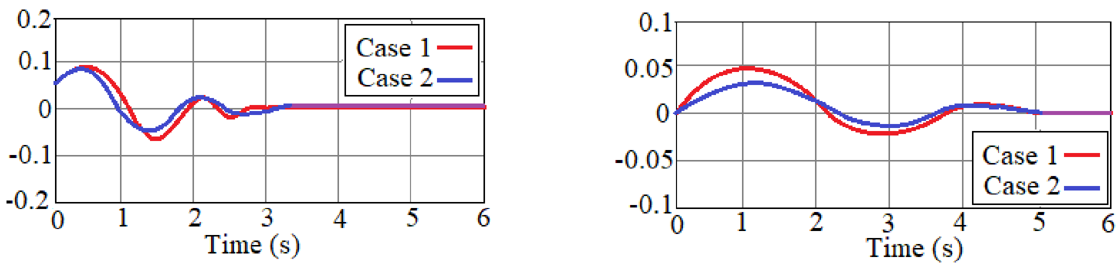
The simulation results at case 1 are to be compared to case 2 to compare system response when changing Kp corresponding slope angle weight α shown in Fig. 6.



(a) (b)

Fig. 6. Comparison of system response between TH1 and TH2: (a)-Angle α; (b)-Angle β

According to results in Fig. 6, decreasing Kp1 causes the overshoot of the tilt angle α to decrease from 0.1 rad to 0.06 rad and the settling time of α dropping from 3 sec to 2.8 sec. The settling time of β is around 5 sec, and this angle does not fluctuate significantly. In contrast, the response also demonstrates the relative independence of the two small SIMO systems.



(a) (b)

Fig. 7. Comparison of system response between TH1 and TH3: (a)-Angle α; (b)-Angle β

According to the results in Fig. 7, decreasing Kp3 causes a small overshoot reduction in the angle β, from 0.09 rads to 0.08 rads, while the settling time remains unchanged.

Thence, changing Kp1, and Kp3, which corresponds to the state variables α and β, also has the same effects as PD tuning in the theory of controlling the SISO system: increasing/decreasing Kp will increase/decrease overshoot, and settling time is longer/shorter.

The tilt angle is stabilized after 3 sec (Fig. 4a), and the reaction wheel move to stabilized position after vibration 0.3 rad (Fig. 6a). Then, the PD controller in Fig. 3a controls well UR when we focus on the model in Fig. 2a. Main purpose of this control is the tilt angle α. Thence, calibration is mainly Kp1 and Kd1 while keeping Kp2 and Kd2.

Angle β is stabilized after 5 sec (Fig. 4b), and the angle of the wheel of UR is also stabilized after the same settling time. The wheel's vibration angle is 0.04 rad (Fig. 6b). Therefore, the PD structure in Figure 5b has successfully controlled the UR when we regard it as a two-wheeled self-balancing robot in Fig. 3b. The main goal in this control is β. So, PD calibration mainly focuses on Kp3, Kd3 and keeping Kp4 and Kd4.

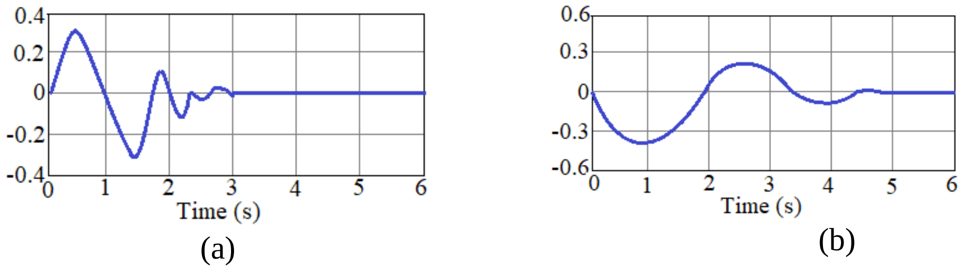


Fig. 8. Comparison of system response in case 1: (a)-Angle α; (b)-Angle β

We tried to change Kd in controllers in Fig. 3, but no exact results were generated. Thence, calibration of Kd is not obtained through this simulation. We mainly use a genetic algorithm in simulation or trial-and-error tests in an experiment to get control parameters. Then, we calibrate Kp for the most focused variables. Increasing Kp makes the system more vibrated. But, if Kp is too small, the control signal is not big enough to control the system to the equilibrium point. When we obtain a good set of PD controllers, the calibration is just around that set.

5. EXPERIMENT

5.1. Experimental Model

Since system (2) has been linearized, simulation can be performed on that linear model. However, because the model is linear instead of nonlinear form, the reliability of the simulation is affected. Then, real experimental results are good to implement for simulation.

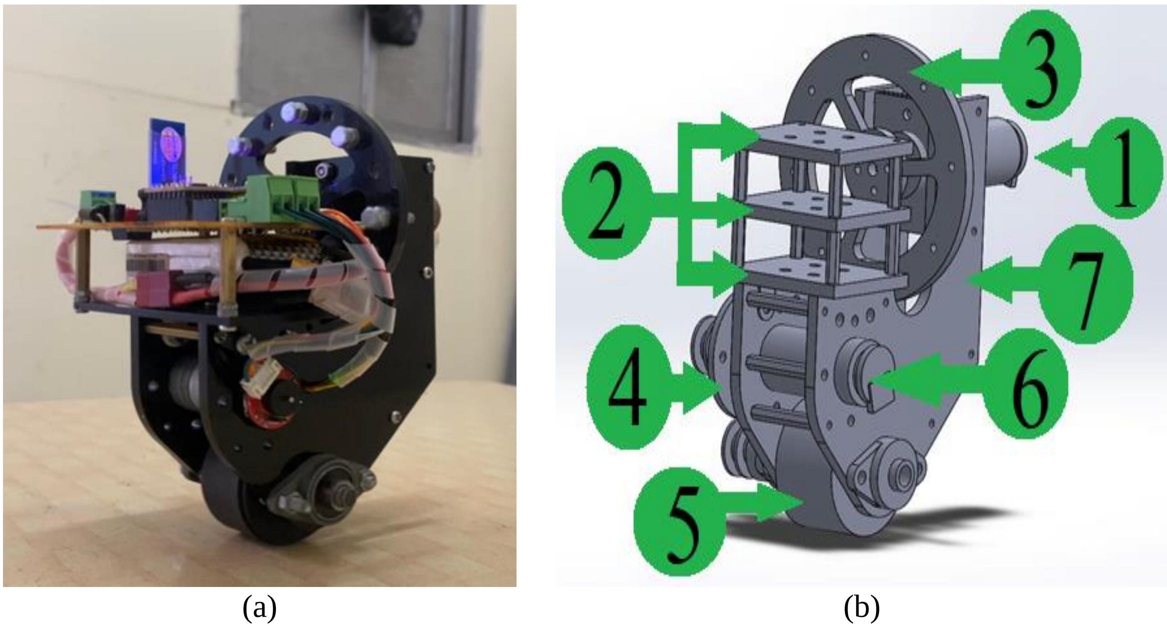


Fig. 9. Experimental model of UR: (a) Real image; (b) SolidWorks image

The following is a list of explanations for the model structure in Fig. 9b:

- DC motor 1 drives the flywheel. An encoder is installed on the motor to track the flywheel's rotation.

- Control variable tray board (Arduino MEGA 2560 board), L298 power board (with 2 channels for 2 motors), 2 MPU6050 sensors for angle of incidence and inclination measurement, HC05 board for signal transmission.
- Flywheel.
- Belt-based gear transmission from the wheel shaft and DC motor shaft 2.
- Wheels.
- Driving wheels powered by a DC motor. An encoder is integrated inside the DC 2 motor to track the wheel's rotation.

Figure 10 below shows the hardware configuration of the UR hardware system control.

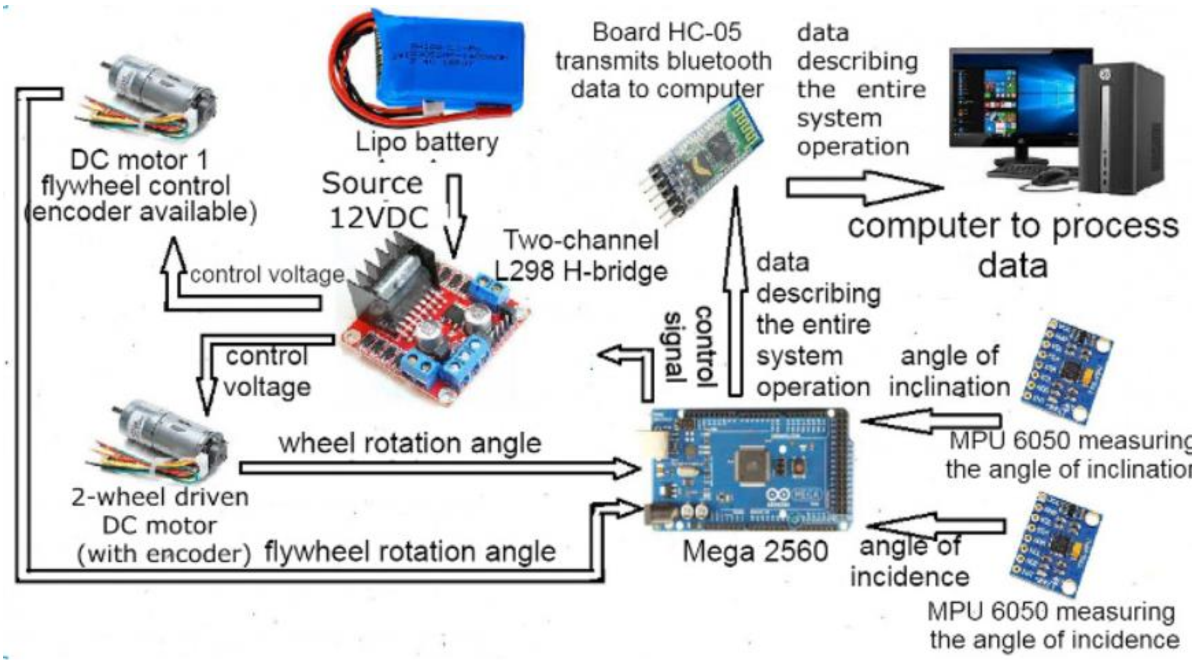


Fig. 10. Configuration of the entire UR system control system

5.2 Experimental Results

A trial-and-error test is utilized due to the unknown system parameters of each model to find PD control parameters in the experiment. Through simulation, Kp is the main component that should be calibrated. Thence, we only focus on Kp in the experiment. Also, angles α and β are variables that will be mainly controlled. These variables correspond to Kp1 and Kp3. Through the trial-and-error test, we selected the following sets of parameters to make the system stable at the equilibrium point:

$$Kp1 = 2; Kd1 = 0.2; Kp2 = 0.06; Kd2 = 4; Kp3 = 4; Kd3 = 1; Kp4 = 6; Kd4 = 0.9 \qquad (6)$$

The experimental results corresponding to the control parameters in (6) are shown in the figures: Fig. 11a, Fig. 12a, Fig. 13a, and Fig. 13b. We can see the PD controller, through trial and error, has successfully controlled the experimental UR. The oscillation of the tilt angle is less than 10 degrees (Fig. 11a), and the incident angle's oscillation is less than 15 degrees (Fig.

12a). Both angles α and β fluctuate around position 0. UR stands upright, not falling. From Fig. 13a, the flywheel swings back and forth around position 0 so that balance of the system does not fall sideways. However, tends to shift to the left (following the dimensional convention of Fig. 3a). This can be explained by the system's center of gravity beings deflected to the right during mechanical fabrication, which is not completely accurate. Correspondingly, in Fig. 13b, the wheel also moves continuously, and UR tends to move forward.

On the basis of the standard set of PD parameters in (6), the parameters Kp1 and Kp3 are adjusted to confirm the examination of the simulation. The failure to correct Kp2 and Kp4 is explained that the designer did not consider the flywheel rotation and wheel rotation angle in balancing the system. The response results of the inclination angle and incidence angle are shown in Fig. 11b and Fig. 12b, respectively:

In Fig. 11, increasing Kp1 from 3 to 5 increases the fluctuation of the tilt angle, but the system remains stable. When increasing Kp1 to 7, the system fluctuates strongly, and UR becomes unstable (no simulation results are shown in this paper). However, if Kp1 is reduced to 1.7, the tilt angle is also unstable. Thus, increasing Kp1 will make the system tilt angle get more oscillation. Increasing it too much will cause instability. The reduction of Kp1 makes the tilt angle stable and less fluctuating. However, the decrease of Kp1 too much makes the control signal not strong enough to stabilize the system in time. Therefore, the selection of Kp1 needs to be chosen appropriately.

In Fig. 12, increasing Kp3 from 10 to 15 increases the fluctuation of the incident angle state variable, but the system remains stable. When increasing Kp3 to 16, the system oscillates strongly and becomes unstable (no simulation results are shown in this paper). However, if Kp3 is reduced to 8.5, the angle of incidence becomes unstable, and the system falls to the front. Thus, increasing Kp3 will make the incident angle easier to oscillate. Increasing it too much will cause instability. Reducing Kp3 makes the angle of incidence more stable, and less oscillating. However, the reduction of Kp3 too much makes the signal not strong enough to stabilize the system in time. Therefore, the selection of Kp3 also needs to be chosen appropriately.

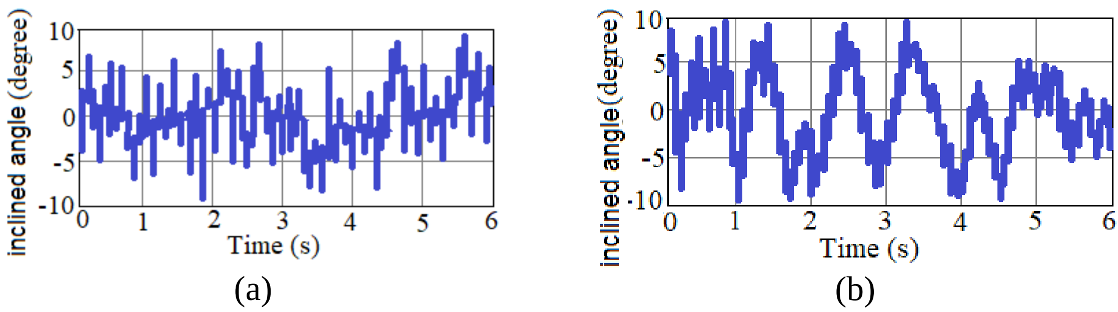


Fig. 11. Response to tilt angle (from MPU6050): (a)-With Kp1=3; (b)-With Kp1=5

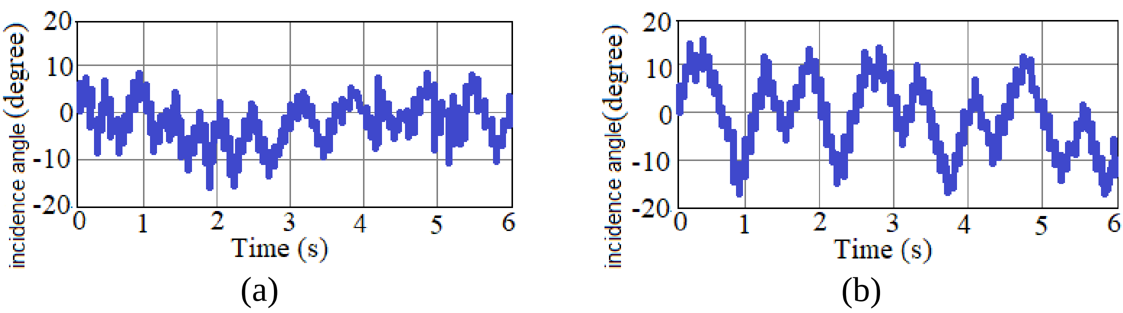


Fig. 12. Incident angle response (from MPU6050): (a)-With Kp3=10; (b)-With Kp3=15

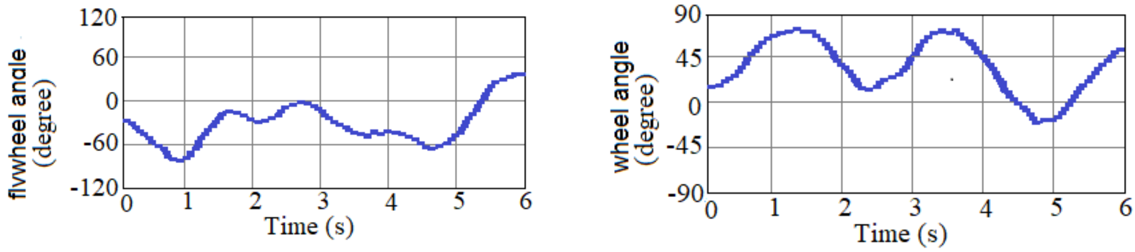


Fig.13. Corresponding response of the UR from the corresponding encoders with the reference parameters in (6): (a)- Flywheel angle p; (b)-Wheel angle

6. CONCLUSION

In this paper, we propose a method of PD control to balance UR not falling. By considering UR as the combination of two other SIMO models: a two-wheeled self-balancing robot and a reaction wheel inverted pendulum, two structures of PID controllers are combined with balancing well UR. Simulation results confirm the ability of our method. Besides, we present an experimental model of UR for real tests. And, Based on this real model, our PD method is proven to work well through experiments. The proportional parameter (Kp) calibration is examined, and the rules of adjustment are confirmed to be the same in both simulation and experiment. Our PD structure can be considered a solution for controlling MIMO under-actuated systems.

REFERENCES

[1] I.-W. Han, J.-W. An, and J.-M. Lee, "Balancing Control of Unicycle Robot," in *Intelligent Autonomous Systems 12: Volume 1 Proceedings of the 12th International Conference IAS-12, held June 26-29, 2012, Jeju Island, Korea*, S. Lee, H. Cho, K.-J. Yoon, and J. Lee Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 663-670.

[2] L. Wei and W. Yao, "Design and implement of LQR controller for a self-balancing unicycle robot," in *2015 IEEE International Conference on Information and Automation*, 8-10 Aug. 2015 2015, pp. 169-173, doi: 10.1109/ICInfA.2015.7279279.

[3] E. K. Ronaghi and S. Seyedtabaii, "Balancing Unicycle Travelling on an Inclined Surface," *Transactions on Machine Intelligence*, 2022, doi: 10.47176/tmi.2022.77.

[4] J. Wu, W. Zhang, and S. Wang, "A Two-Wheeled Self-Balancing Robot with the Fuzzy PD Control Method," *Mathematical Problems in Engineering*, vol. 2012, p. 469491, 2012/11/26 2012, doi: 10.1155/2012/469491.

[5] J. F. D. Vries, "Redesign & Implementation of a Moment Exchange Unicycle Robot," University of Twente, 2018.

[6] D. E. Zilio and V. B. Peccin, "Multivariable Control In Industrial Processes: Comparison Of MIMO LQR And Multi-SISO PID Techniques Applied To An Industrial Plant In Scale," in *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, 12-14 Nov. 2018 2018, pp. 1165-1172, doi: 10.1109/INDUSCON.2018.8627253.

[7] F. C. Paulescu, I. Szeidert, I. Filip, and C. Vasar, "Two-Wheeled Self-Balancing Robot," in *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 19-21 May 2021 2021, pp. 000033-000038, doi: 10.1109/SACI51354.2021.9465568.

[8] G. Belascuen and N. Aguilar, "Design, Modeling and Control of a Reaction Wheel Balanced Inverted Pendulum," in *2018 IEEE Biennial Congress of Argentina (ARGENCON)*, 6-8 June 2018 2018, pp. 1-9, doi: 10.1109/ARGENCON.2018.8646093.

[9] A. T. Ali, A. M. O. Mohamedy, A. S. A. Salimz, E. A. O. M. El-Aminx, and O. M. K. Ahmed, "Design and Implementation of Two-Wheeled Self-Balancing Robot Using PID Controller," in *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 26 Feb.-1 March 2021 2021, pp. 1-5, doi: 10.1109/ICCCEEE49695.2021.9429579.

[10] H. W. Kim, J. W. An, H. d. Yoo, and J. M. Lee, "Balancing control of bicycle robot using PID control," in *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, 20-23 Oct. 2013 2013, pp. 145-147, doi: 10.1109/ICCAS.2013.6703879.