UNIJOURN

International Journal of Sustainable Transportation Technology
Vol. 3, No. 1, April 2020, 12-19

Design and Simulation of Two-Wheeled Balancing Mobile Robot with PID Controller

Vicky Mudeng[1*], Barokatun Hassanah[1], Yun Tonce Kusuma Priyanto[1,2], Okcy Saputra[1]

[1]Department of Electrical Engineering, Institut Teknologi Kalimantan, Indonesia
[2]National Center for Sustainable Transportation Technology, Indonesia
**Email: mudengvicky@lecturer.itk.ac.id*

Abstract

Mobile transportation robots using two wheels have now been investigated. The work within this study is to design and simulate two-wheeled robots, thus it can maintain its balance. Many control methods are used to determine satisfactory control optimization, therefore a proper response is obtained by sensor recitation corresponding with the reaction of a Direct Current (DC) motor. Recently, two-wheeled transportation robot is a Segway model. In this study, we apply a Proportional Integral Derivative (PID) controller as a control system in a self-balancing robot with a working principle is similar to an inverted pendulum. In the next study, the PID controller and the whole system are applied in the microcontroller board. The angular velocity of two DC motors used as a plant can be adjusted by Pulse Width Modulation (PWM) through a motor driver. An Inertial Measurement Unit (IMU) sensor is utilized to detect the angular acceleration and angular velocity of the self-balancing robot. The phase design is constructed by planning the robot dimension, mechanical system, and an electronic system. Particularly, this study performs mathematical modeling of the robot system to obtain the transfer function. In addition, we simulate the PID parameter with multiplication between the basic parameter and several fixed constants. The simulation results indicate that the robot can maintain its balance and remains perpendicularly stable for balancing itself.

Keywords

Mobile transportation; Proportional integral derivative; Robot; Self-balancing; Segway

1 Introduction

The development of robotics technology has made the quality of human life even higher. Nowadays, it has been able to improve both the quality and quantity, in terms of production, medicine, education, and competition [1]. Moreover, it has reached a short distance transportation without pollution due to the usage of an electricity source, for instance, a two-wheeled robot has been utilized as a means of transportation, namely Segway [2, 3]. It is expected to be an alternative solution for people who expect to have a practical vehicle that can be used for personal transporter [4]. One of them is a self-balancing two-wheeled robot [5–8].

Two-wheeled balancing robot is a mobile robot with two wheels on the right and left sides and it will not be balanced without a controller. A self-balancing robot is an application of an inverted pendulum model that is placed on a wheeled train with the aim is to keep the robot upright without outside control. The balancing system in a two-wheeled robot requires a sufficient and reliable control method. These are required to maintain the position of the robot in a state which perpendicular to the surface of the earth without the necessity of other controllers from outside [9].

Previous research discussed the application of Kalman filtering and Proportional Integral Derivative (PID) control for an inverted pendulum controller. This research developed a prototype design of a self-balancing robot using a Kalman filtering algorithm and PID control. The Kalman filtering algorithm was employed to reduce or eliminate several error signals from the output of the sensor [10]. The PID control algorithm was used to regulate the forward, backward, rotating, and stabilizing perpendicular movement to the flat plane, also to rotate the direction of the robot [11-15].

Other studies presented self-balancing robots with comparison of control methods, such as Proportional Derivative (PD), Proportional Integral (PI), and PID. This study aims to explain the design and building of a self-balancing robot with various control methods, thus the robot can balance perpendicular to the earth surface in a flat plane area. The microcontroller used was AVR ATMega 128, and the sensor was MPU-6050 module. Controllers were deployed to determine the velocity and

Received February 4, 2020, Revised March 16, 2020, Accepted for publication April 10, 2020

direction of Direct Current (DC) motor rotating as a plant, based on the tilt angle of the robot body to the surface of the flat plane. This self-balancing robot can maintain the position corresponding with the earth surface in a flat plane [16].

The objective of this study is to design and simulate a Two-Wheeled Balancing Mobile Robot (TWBMR) model with the PID control algorithm as a controller in determining the magnitude and direction of rotation for the DC motor. A DC motor was as a plant, hence TWBMR can maintain its position perpendicular to the earth surface in a flat plane. To ensure the design, we emphasized the simulation of PID controller with a transfer function of the TWBMR model. Then, we tried to conduct multiplication between the calculated PID parameter with a set of constants.

According to the above introduction, the organization of this paper is as follows. In Section 2, we describe the mathematical modeling of the robot system; also, we state the parameters of TWBMR. Subsequently, the robot modeling system with type 1 and 2 of the Ziegler-Nichols PID controller is demonstrated in Section 3, where discussions are provided as well. Section 4 presents simulation results with type 1 and 2 of the Ziegler-Nichols PID controller. Finally, some remarks and conclusions of the study are stated in Section 5.

2 Mathematical Modeling of Robot System

The transfer function is obtained through mathematical calculations of the self-balancing robot mechanical system. The mechanical concept of a TWBMR is the same as the inverted pendulum concept. The pendulum will be fallen when there is an absence of balancing force. Based on the concept, it can be concluded that the system is unstable. In contrast, the pendulum position is normal or not reversed, thus it is stable. Figure 1 depicts the model for an inverted pendulum with θ is tilt angle, F is force given to the model, M denotes model mass, x and y are axis of vertical and horizontal, respectively. In the implementation, DC motors will provide the appropriate velocity, thus the robot remains balance. Without the appropriate velocity, the robot will befallen.

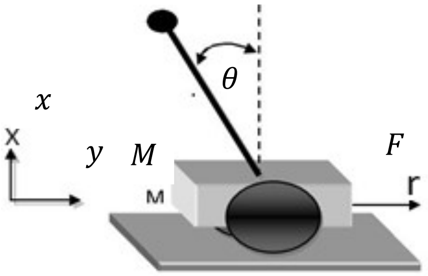

Figure 1 Inverted pendulum model

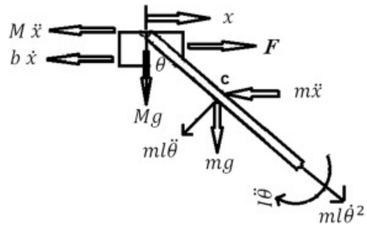

Figure 2 Inverted pendulum free body and inertia force diagram at the stable equilibrium position

Table 1 Parameters of TWBMR

Variable	Description	Value
M	Mass of the cart	0.3 kg
m	Mass of pendulum rod	0.2 kg
b	Viscous friction coefficient	0.1 N/m/sec
l	One half pendulum rod length	0.15 m
I	Inertia moment pendulum rod	0.0015 kg.m^2
F	Applied force to the cart	kg.m/s^2
g	Gravity	m/s^2
θ	Pendulum angle	rad

The dynamic equation for TWBMR from Newton II law is applied. Figure 2 shows the Free Body Diagram (FBD) of the mechanism. While the slope of the pendulum with several angles to complete the two-component of forces along with the horizontal and vertical directions. Table 1 explains the parameters for TWBMR. According to Figure 2, it can be obtained the mathematical system modeling equation for self-balancing robots, as follows [17]

$$\Sigma f_x = 0$$
$$F - b\dot{x} - M\dot{x} - m\ddot{x} - ml\ddot{\theta}cos\theta + ml\dot{\theta}^2 sin\theta = 0 \tag{1}$$

$$\Sigma f_y = 0$$
$$Mg + mg - ml\ddot{\theta}sin\theta - ml\dot{\theta}^2 cos\theta = 0 \tag{2}$$

$$\Sigma M_A = 0 \tag{3}$$
$$mgl\ sin\theta + I\ddot{\theta} + ml^2\ddot{\theta} + ml\ddot{x}cos\theta = 0,$$

Equations (1), (2), and (3) are combined, hence the force equation is obtained along the horizontal and vertical directions,

$$(I + ml^2)\ \theta + mgl\ sin\theta = -ml\ddot{x}cos\theta \tag{4}$$

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta = F, \tag{5}$$

Equations (4) and (5) are two linear equations of the transfer function, where $q = \pi$. With assumption $\theta = \pi + \phi$,

$$cos\theta = -1 \tag{6}$$

$$sin\theta = -\phi \tag{7}$$

$$\frac{d^2}{dt^2} = 0, \tag{8}$$

After processing the equations (6), (7), and (8) approach to non-linear, then we obtained two variations of motion equations. The U value represents the input,

$$(I + ml)^2\,\ddot{\Phi} - mgl\,\Phi = ml\ddot{x} \tag{9}$$

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\Phi} = U, \tag{10}$$

By applying Laplace transform on equations (9) and (10),

$$(I + ml^2)\ddot{\Phi}(s)s^2 - mgl\,\Phi(s) \ = mlX(s)s^2 \tag{11}$$

$$(M + m)X(s)s^2 + bX(s)s + ml\Phi(s)s^2 = U(s), \tag{12}$$

we yielded the transfer function for TWBMR from the equations (11) and (12).

$$\frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}\,s}{s^3 + \frac{b(I+ml^2)}{q}s^2 - \frac{(M+m)mgl}{q}s - \frac{bmgl}{q}}, \tag{13}$$

where $q = [(M + m)(I + ml^2) - (ml)^2]$.

The parameters in Table 1 are substituted in equation (13) by replacing the q value in equation (13), we obtained the transfer function of TWBMR.

$$\frac{\Phi(s)}{U(s)} = \frac{14{,}286s}{s^3 + 0.286s^2 - 70s - 0.0294} \tag{14}$$

3 PID Controller

Each deficiency and strengthen of P, I and D controllers can cover each other by combining them in parallel into the PID controller. The combination of proportional, integral and derivative controls in the PID control system has a specific purpose. Proportional control excels at fast rise time, while integral control can eliminate the steady-state error. Moreover, derivative control can reduce overshoot. When we combined them, we get the result of control with error-eliminating properties, reducing in rise time, settling time, and overshoot [17]. Table 2 shows the PID controller characteristics.

3.1 Simulation design with type 1 of Ziegler-Nichols PID controller

After obtaining the transfer function from the systematic modeling of self-balancing robots. Afterward, we simulated the function in equation (14) using Simulink MATLAB. It is aimed to perform the system response from the transfer function of TWBMR. The diagram block of the system simulation test using the PID Ziegler-Nichols type 1 method is shown in Figure 3, which consists of the unit step, transfer function, and output. The value of unit step was given 90, which means that the response system will be in steady-state at a value of 90°. The value of the transfer function block was given according to equation (14) due to the response system value to be tested for validating the stable condition. The output/scope will display the results of the response system wave from the tests carried out by the PID Ziegler-Nichols type 1 method [17].

Table 2 Characteristics of PID parameters

Parameter	Rise Time	Overshoot	Settling Time	Error Steady State
K_p	Decrease	Increase	Decrease by Variation	Decrease
K_i	Decrease	Increase	Increase	Elimination
K_d	Decrease by Variation	Decrease	Decrease	Decrease by Variation

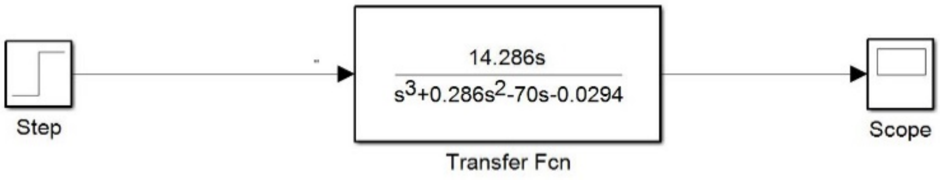

Figure 3 TWBMR system diagram without PID controller

3.2 Simulation design with type 2 of Ziegler-Nichols PID controller

The PID control method is used to improve the system response to self-balancing robots. The response system obtained without using the PID control system was still far from the desired response and even the robot did not reach the specified setpoint. The first thing to do is utilized PID control method to calculate the values of

K_p, K_i, and K_d as parameters of the PID controller. The determination of PID parameters is based on the reaction of the closed-loop system in the transfer function of self-balancing robots. A K_p variable is added first to the system, thus the closed-loop transfer function is determined by adding the proportional gain (K_p) [8], as follows

$$\frac{C(s)}{R(s)} = \frac{14{,}286s}{s^3 + 0.286s^2 - 70s - 0.0294} \tag{15}$$

$$\frac{C(s)}{R(s)} = \frac{K_p \frac{14.286s}{s^3+0.286s^2-70s-0.0294}}{1 + K_p\left(\frac{14.286s}{s^3+0.286s^2-70s-0.0294}\right)} \tag{16}$$

$$\frac{C(s)}{R(s)} = \frac{K_p 14{,}286}{K_p\ 14.286 + (s + 8.51)\ (s - 8.22)\ (s + 0.0004)} \tag{17}$$

$$\frac{C(s)}{R(s)} = \frac{K_P\ (14{,}286)}{K_p\ 14.286 + s^3 + 0.286s^2 - 70s - 0.0294} \tag{18}$$

The stability of a system can be seen from the location of the pole system in the field s, if the poles of the system are located to the left of the field s, then the system is stable. To find out the location of the poles in a system, hence we used the stability of routh. The first column in routh stability shows the polar location of a system, if the variable in the column is positive then it can be ascertained that a system has a pole on the left side of the field s. To get K_p value to meet the routh stability criteria, it can be determined by referring to the transfer function that has been obtained in equation (18),

$$s^3 \rightarrow 1 \qquad -70$$

$$s^2 \rightarrow 0.286 \qquad -0.0294 + 14.286\ K_p$$

$$s^1 \rightarrow \frac{-20.02 + (0.0294 - 14.286K_p)}{0.286}$$

$$s^0 \rightarrow -0.0294 + 14.286K_p.$$

Then, s^1 can be analyzed as follows

$$\frac{-20.02 + (0.0294 - 14.286K_p)}{0.286} \geq 0$$

$$-70 + 0.103 - 49.95K_p \ \geq 0$$

$$-69.897 - 49.95K_p \ \geq 0$$

$$K_p \ \geq -1.399 \tag{19}$$

Whereas, for s^0, as follows

$$-0.0294 + 14.286\ K_P \ \ \geq 0 \tag{20}$$

$$K_p \geq 0.00205, \tag{21}$$

Therefore, we acquired the range of K_p for TWBMR as follows,

$$-1{,}399 \geq K_P \geq 0{,}00205 \tag{22}$$

K_{cr} and P_{cr} were yielded by substituting $j\omega$ in the variable s for denominator of equation (18). Denominator function over the closed-loop system commonly referred to as the equation of system characteristics. System characteristic equations with K_p parameter on self-balancing robots is as follows,

$$s^3 + 0.286s^2 + 70s - 0.0294 + 14.286K_p \ = 0 \tag{23}$$

$$-j\omega^3 - 0.286\ \omega^2 + 70j\omega - 0.0294 + 14.286K_p = \ 0 \tag{24}$$

$$\left(-0.286\omega^2 - 0.0294 + 14.286K_p\right) + j(-\omega^3 + 70\omega) = 0. \tag{25}$$

From equation (25), we obtained two parts, namely the real and imaginary parts. The imaginary part will be used to get the value of ω, while the real part is used to get the K_p value,

$$(-\ \omega^3 + 70\omega) = 0 \tag{26}$$

$$\omega = 8.366. \tag{27}$$

According to equation (27), we yielded P_{cr}

$$P_{cr} = \frac{2\pi}{\omega} = \frac{2\pi}{8.366} = 0.751\ second, \tag{28}$$

The K_{cr} value was obtained by using the real part by substituting the value ω.

$$\left(-0.286\omega^2 - 0.0294 + 14.286K_p\right) = 0 \tag{29}$$

$$K_p = 1.3999, \tag{30}$$

$K_p = K_{cr} = 1.3999$. With K_{cr} and P_{cr} obtained values, thus, the value of the Ziegler-Nichols type 2 PID parameters for TWBMR is presented in Table 3.

After the parameters of the K_p, K_i, and K_d values are obtained, a simulation can be carried out by adding the PID controller to the transfer function of the self-balancing robot, as shown in Figure 4.

Table 3 Parameters of the PID Ziegler-Nichols for TWBMR

Parameter	PID Value
K_p	4.0602
$T_i = K_i$	4.4100
$T_d = K_d$	1.1025

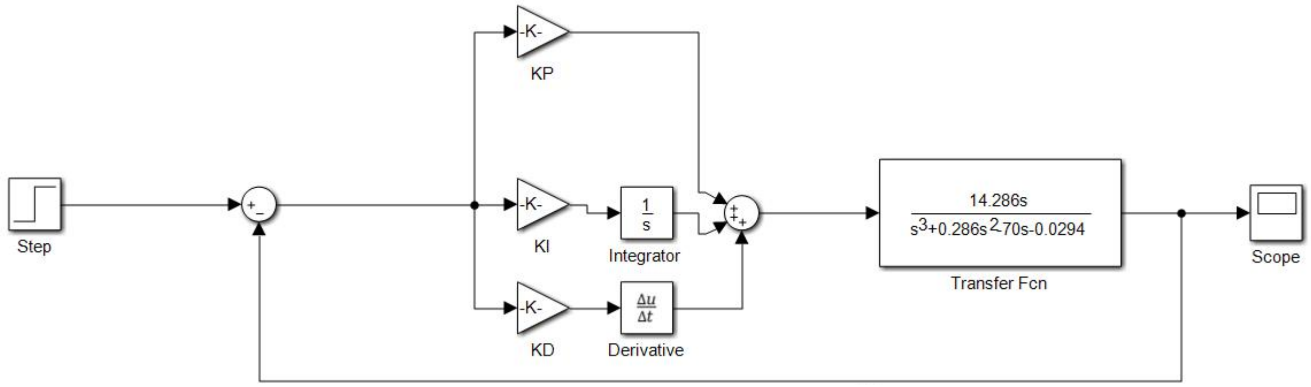

Figure 4 TWBMR system diagram with PID controller

4 Result and Discussion

Based on the diagram block in Figure 3, we presented the response from the self-balancing robot transfer function which was tested without using a controller by performing a system simulation on Simulink MATLAB. From the simulation results, the system response graph was obtained as shown in Figure 5.

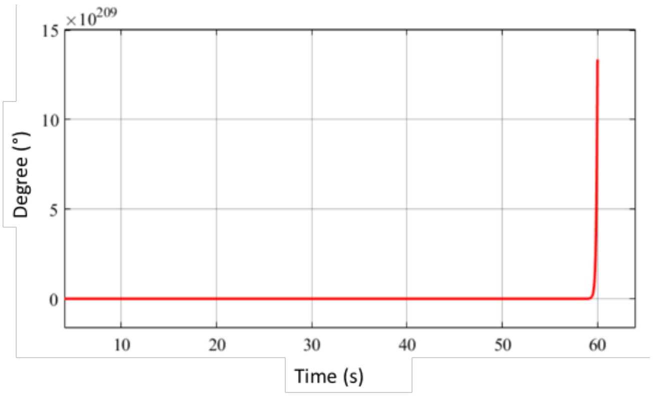

Figure 5 Response for TWBMR system without PID controller

Figure 5 shows the response system generated from the self-balancing robot simulation when using the PID Zigler-Nichols type 1 method. Besides, it shows a response system, despite it does not form the s curve due to the usage of an opened-loop system. Even, if the system directly implemented to the TWBMR, thus it will fall concerning its balanced condition. Therefore, to improve the response system for self-balancing robot, it is necessary to add a control system. The control system or controller that will be added to the self-balancing robot system is the PID controller.

4.1 PID system simulation with the type 2 Ziegler-Nichols method

Based on the diagram block in Figure 4 to see the response of the transfer function for TWBMR, and using parameters in Table 3, the response generated with the PID control can be seen in Figure 6.

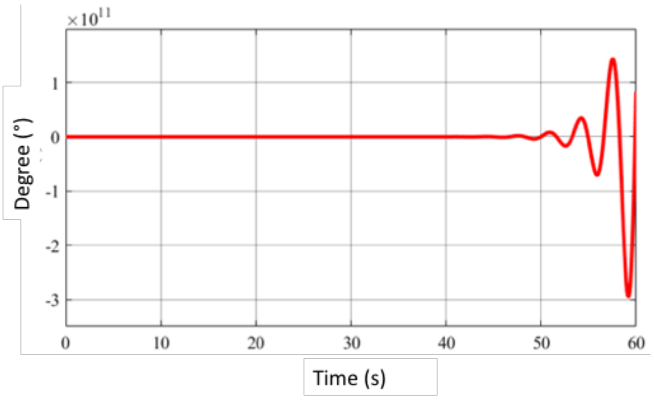

Figure 6 Response system for TWBMR system diagram with PID controller

From Figure 6, the response system is obtained which at the beginning of the robot will be stable in a few seconds. When at 52 seconds the robot starts oscillating left and right. In this state, the robot will try to balance its position by looking for the set point value that has been determined, namely 1. Testing the PID simulation with K_p, K_i, and K_d values by calculating Ziegler-Nichols type 2 calculation can balance the robot, however only just a moment and over time the robot will fall due to its failure to maintain the balance.

After testing the PID Ziegler-Nichols type 2, the result was not as expected. Therefore, the next test was conducted by adding two times for each value of K_p, K_i, and K_d. This addition was done to improve the response of the system to suit what desired was and refer to the following characteristics in Table 2. Figure 7 shows the result of the simulation for the PID test with the addition of two multiplied by the parameters in Table 3.

Figure 7 shows that the overshoot value is still fairly high, and the rise time is moderately fast, but for the value of settling time is still quite long which is about 8 seconds. These conditions are caused by the function of each used controller, such as the function of the P controller is to speed up the rise time, but it can increase overshoot. The function of the controller I can reduce errors and speed up rise time, but it can increase settling time and overshoot.

While the function of the D controller is to provide an overshoot damping effect and reduce settling time. Besides, in Figure 8 when we simulated the system with K_P times 2 and maintain the values of K_i and K_d, as in Table 3. As can be seen, the overshoot value is incredibly high, but the rise time is pretty fast and the settling time is fairly fast from the previous test.

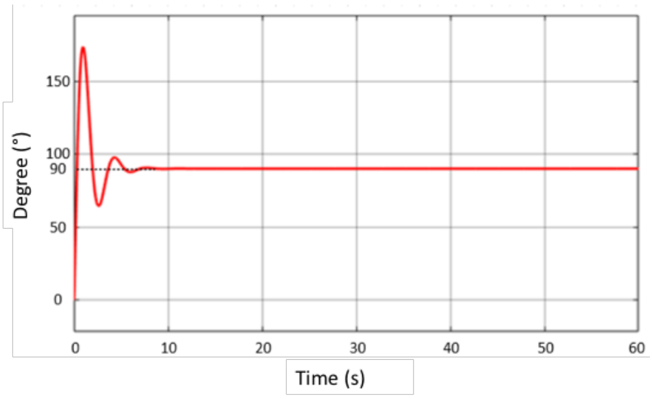

Figure 7 TWBMR system without PID controller with Parameters of the PID Ziegler-Nichols using doubled parameter values

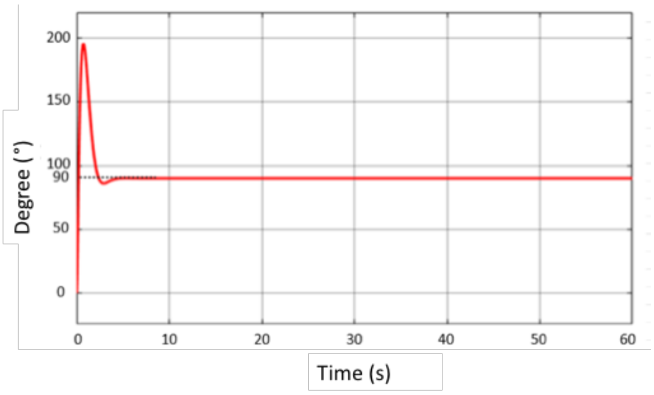

Figure 8 Response system for TWBMR system diagram with PID controller using doubled K_p value

Additionally, in Figure 9, the overshoot value is extremely high and the settling time is quite long comparing the test in Figure 7, but the rise time is quite fast.

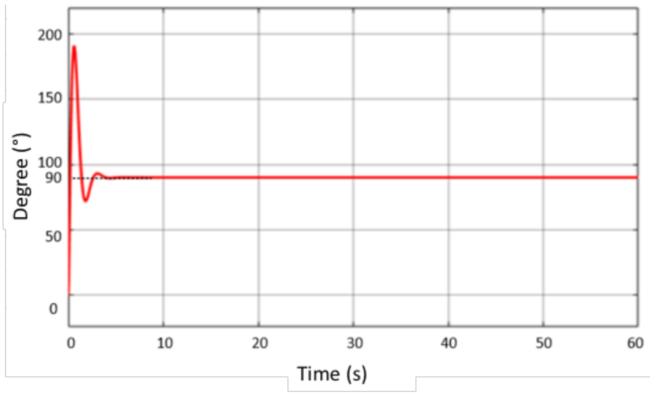

Figure 9 Response system for TWBMR system diagram with PID controller using doubled K_p and K_i values

We multiplied 2 to the value of K_p and K_d. The values of settling time and rise time are relatively extended, but the overshoot value is quite fast if we compare with Figure 8.

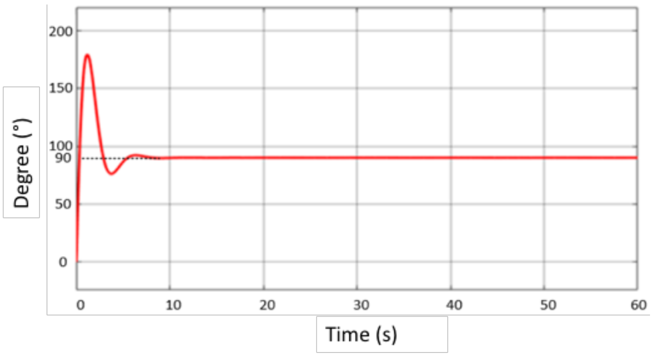

Figure 10 Response system for TWBMR system diagram with PID controller using doubled K_p and K_d values

Not only conducted multiplication with 2 for all parameters in Table 3 but also, we tried to simulate with multiplication with 10 for the values of K_p, K_i, and K_d, as shown in Figure 10. The overshoot value is fairly small from the two previous tests and the rise time value is obtained, as well as the value of settling time is similar as in Figure 7. Also, we simulated K_p, K_i, and K_d times 15, as shown in Figure 11. The overshoot value is relatively small and the rise time value is obtained with the value of settling time is faster.

Furthermore, we conducted multiplication between 20 and K_p, K_i, and $K_d$values, as shown in Figure 13. The overshoot value is small and the rise time value is obtained. In addition, the settling time is faster. These results almost similar to the result in Figure 12.

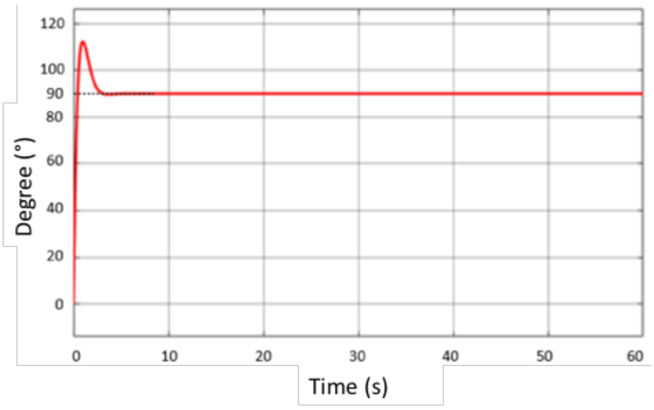

Figure 11 Response system for TWBMR system diagram with a PID controller using parameters value multiplied by 10

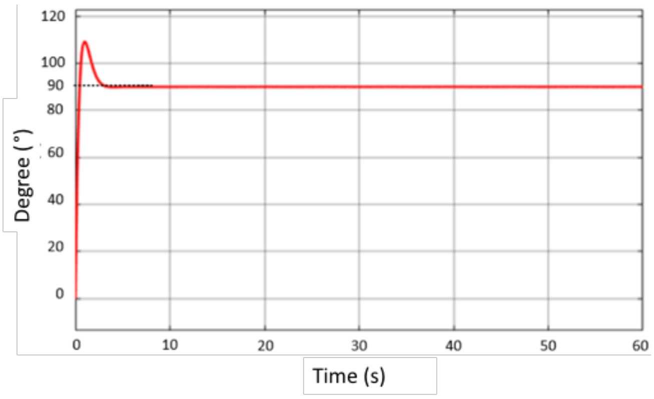

Figure 12 Response system for TWBMR system diagram with a PID controller using parameters value multiplied by 15

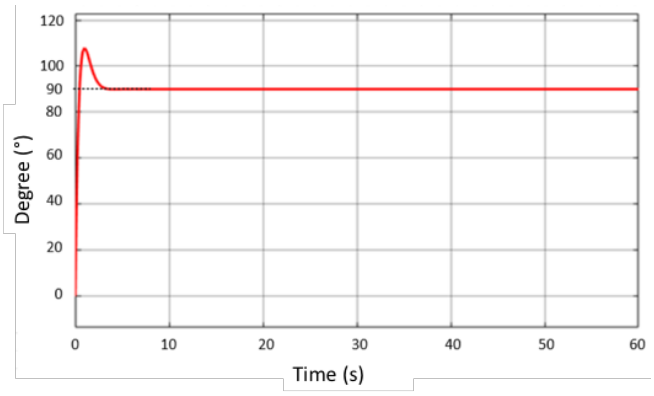

Figure 13 Response system for TWBMR system diagram with a PID controller using parameters value multiplied by 20

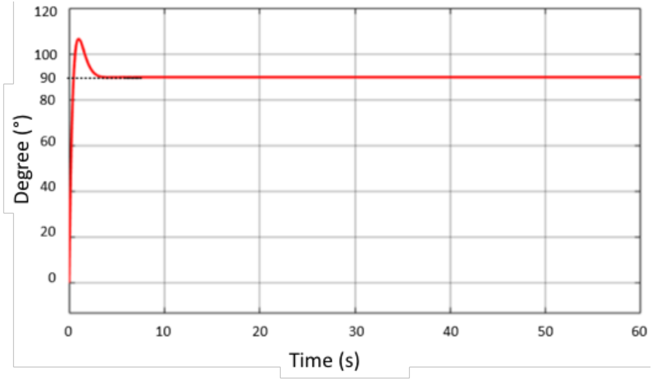

Figure 14 Response system for TWBMR system diagram with a PID controller using parameters value multiplied by 25

To verify the system with the expected result, therefore the last testing was multiplying 25 for the K_p, K_i, and K_d values, as shown in Figure 14. The overshoot value is moderately small and even approaches 90° or setpoint and the rise time value is obtained, as well as the settling time value is faster than all the tests that have been done.

Table 4 shows the simulating response systems for the whole parameters of PID. By looking at the response of the system when the values of K_p, K_i, and K_d are multiplied by 25, we can verify that the TWBMR simulation is suitable for implementation with appropriate PID controller parameters.

The maximum overshoot value smaller than the previous tests even close to the specified setpoint value. The rise time is quite fast from previous tests and the value of steady-state error and settling time are quite small compared to all tests that have been done. These are because the function of each controller used, such as the function of controller P speeds up the rise time, but it can add overshoot.

The function of the controller I can reduce error and speed up rise time, but it can increase settling time and overshoot. The function of controller D is to provide an overshoot and reduce settling time. Hence, the best PID simulation value is at the K_p, K_i, and K_d values are multiplied by 25.

Table 4 Response systems for testing simulation

PID Controller	Maximum Overshoot (%)	Rise Time (s)	Error Steady State (%)	Settling Time (s)
2 times K_p, K_i, and K_d	92.6	0.316	0	8.879
2 times K_p	117.2	0.201	0	5.259
2 times K_p and K_i	111.5	0.259	0	5.482
2 times K_p and K_d	99.2	0.316	0	8.764
10 times K_p, K_i, and K_d	24.7	0.431	0	4.741
15 times K_p, K_i, and K_d	20.6	0.489	0	4.224
20 times K_p, K_i, and K_d	19.6	0.431	0	3.882
25 times K_p, K_i, and K_d	18.6	0.374	0	3.649

5 Conclusion

The design value of the PID controller has been obtained by validating the transfer function first and then simulated it using Simulink MATLAB simulation software. The results obtained by Simulink MATLAB simulation with the best system simulation test value for balancing robots are when the values of K_p, K_i, and K_d are multiplied by 25 for each value. These results ensure that TWBMR possible to implement to be a means of transportation or personal transporter.

Acknowledgment

This research was supported by USAID through Sustainable Higher Education Research Alliances (SHERA) Program – Centre for Collaborative (CCR) National Center for Sustainable Transportation Technology (NCSTT) and funded by the Ministry of Research, Technology and Higher Education of the Republic of Indonesia through grant 007/SP2H/LT/DRPM/2019.

References

[1] H. S. Juang and K. Y. LumLum, "Design and Control of a Two-Wheel Self-Balancing Robot using The Arduino Microcontroller Board," in *10th IEEE International Conference on Control and Automation*, 2013, pp. 634–639.

[2] J. Wu, W. Zhang, and S. Wang, "A Two-Wheeled Self-Balancing Robot with The Fuzzy PD Control Method," *Math. Probl. Eng.*, vol. 469491, pp. 1–13, 2012.

[3] S. Haddout, "Nonlinear Reduced Dynamics Modelling and Simulation Of Two-Wheeled Self-Balancing Mobile Robot: SEGWAY System," *Syst. Sci. Control. Eng.*, vol. 6, no. 1, pp. 1–11, 2018.

[4] K. Kankhunthod, V. Kongratana, A. Numsomran, and V. Tipsuwanporn, "Self-Balancing Robot Control using Fractional-Order PID Controller," in *International Multi Conference of Engineers and Computer Scientists*, 2019.

[5] M. I. Ali and M. Hossen, "A Two-Wheeled Self-Balancing Robot With Dynamics Model," in *4th International Conference on Advances in Electrical Engineering*, 2017, pp. 271–275.

[6] T. M. Ravikumar and R. Saravanan, "Reduction of Odometry Error in a Two Wheeled Differential Drive Robot," *Int. J. Eng.*, vol. 27, no. 3, pp. 359–366, 2014.

[7] J. H. Park and B. K. Cho, "Development of a Self-Balancing Robot with a Control Moment Gyroscope," *Int. J. Adv. Robot Syst.*, vol. 15, no. 2, pp. 1–11, 2018.

[8] S. Kim and S. Kwon, "Dynamic Modeling of a Two-Wheeled Inverted Pendulum Balancing Mobile Robot," *Int. J. Control Autom.*, vol. 13, no. 4, pp. 926–933, 2015.

[9] K. Liu, M. Bai, and Y. Ni, "Two-Wheel Self-Balanced Car Based on Kalman Filtering And PID Algorithm," in *The 18th International Conference on Industrial Engineering and Engineering Management*, 2011, pp. 281–285.

[10] I. Faruqi, M. B. Waluya, Y. Y. Nazaruddin and T. A. Tamba, "Train Localization using Unscented Kalman Filter - Based Sensor Fusion," *Int. J. Sustain. Transp. Technol.,* vol. 1, no. 2, pp. 35–41, 2018.

[11] F. Sun, Z. Yu, and H. Yang, "A Design For Two-Wheeled Self-Balancing Robot Based on Kalman Filter And LQR," in *The International Conference on Mechatronics and Control*, 2014, pp. 612–616.

[12] C. B. Kadu and C. Y. Patil, "Design and Implementation of Stable PID Controller for Interacting Level Control System," *Procedia Comput. Sci.*, vol. 79, pp. 737–746, 2016.

[13] S. Saxena and Y. V. Hote, "Internal Model Control Based PID Tuning Using First-Order Filter," *Int. J. Control Autom.*, vol. 15, pp. 145–149, 2017.

[14] A. Y. Jaen-Cuellar, R. Romero-Troncoso, L. Morales-Velazquez, and R. A. Osornio-Rios, "PID-Controller Tuning Optimization with Genetic Algorithms in Servo Systems," *Int. J. Adv. Robot Syst.*, vol. 10, no. 324, pp. 1–14, 2013.

[15] V. Chopra, S. K. Singla, and L. Dewan, "Comparative Analysis of Tuning a PID Controller using Intelligent Methods," *Acta Polytech. Hung.*, vol. 11, no. 8, pp. 235–249, 2014.

[16] B. Y. Suprapto, D. Amri, and S. Dwijayanti, "Comparison of Control Methods PD, PI, and PID on Two Wheeled Self Balancing Robot," in *the International Conference on Electrical Engineering, Computer Science and Informatics*, 2014, pp. 67–71.

[17] K. Ogata, *Modern control engineering 5th Edition*. Prentice Hall, 2010.