

# TWO-WHEELED BALANCING ROBOT WITH ANDROID NAVIGATION SYSTEM

<sup>1</sup>BAKTI DWI WALUYO, <sup>2</sup>DADANG MULYANA, <sup>3</sup>BAHARUDDIN, <sup>4</sup>ARIF RAHMAN,

<sup>5</sup>MUHAMMAD AULIA RAHMAN SEMBIRING

<sup>1,2,3,4,5</sup>Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Medan, Indonesia

E-mail: <sup>1</sup>bakti\_dw@unimed.ac.id, <sup>2</sup>dadang@unimed.ac.id, <sup>3</sup>baharuddin@unimed.ac.id,  
<sup>4</sup>arif81@gmail.com, <sup>5</sup>marsembiring@unimed.ac.id

## ABSTRACT

The two-wheeled balancing robot is a robot that can move with two wheels on the left and the right. However, in order to maintain balance, the robot needs to use both wheels to move. We, therefore, need control to move the two-wheel robot so that it can stand in balance. This system has two inputs, namely accelerometers, used to measure angular acceleration ( $m/s^2$ ) and gyroscopes for measuring angular velocity ( $rad/s$ ). The accelerometers and gyroscope values were calculated using the complementary filter method to obtain the angle values. The angle obtained is then compared with the setpoint, which is  $0^\circ$ . The difference between the setpoint and the complementary filter angle is processed using the Proportional Integral Derivative (PID) control method. The PID control process results are used to regulate the rotation of the wheel drive motor in the robot. The direction of the wheel drive motor rotation will go forward if the complementary filter angle is less than zero and reverse if it is more than zero. Based on the tests that have been done, the balancing robot can withstand an angle range of  $-1.5^\circ$  to  $1.5^\circ$ . While the PID constant value is  $K_p = 1.5$ ,  $K_i = 0.2$ , and  $K_d = 0.05$  and the coefficient value on the complementary filter algorithm is  $\alpha = 0.96$ . The two-wheeled balance robot can be operated with an Android smartphone via Bluetooth properly and can move in balance by lifting a maximum load of 40 Kilograms.

**Keywords:** *Balancing Robot, Complementary Filter, Android Smartphone, PID*

## 1. INTRODUCTION

Research on two-wheeled balancing robot (TWBR) has attracted the attention of robotics practitioners, and this has led to its development towards the commercial direction [1][2]. Research on TWBR continues because of the dynamics of instability, disturbance, and uncertainty of parameters that appear in the system [3][4]. TWBR is a suitable tool for testing control theory and control methods [5]. TWBR modeling and control methods from 2000 to 2012 were reviewed in [6], which provides a detailed description of the various TWBR configurations to handle challenges; for example, the robot can avoid obstacles and walk on uneven surfaces [7].

Many control system algorithm models have been applied to TWBR. Among them is "Joe", a DSP-based wheel balancing robot developed by Grasser; it achieves good mobility through decoupling movements and simplified control but has low interference resistance [1]. Miasa designed the self-balancing robot "Equibot," but it is only suitable for moving on flat ground and can fall on

uneven ground with high road conditions [8]. Wu applied fuzzy to robots, but this method is expensive and time-consuming due to fuzzy rules [9]. Artificial neural networks have also been applied to TWBR, and the system slows down as all weights are updated during the learning process [10][11].

TWBR is a robot that can balance itself with two wheels placed on the right and left. TWBR is a development of the inverted pendulum model, which maintains the robot's balance at an angle of  $0^\circ$  [13]. It uses a gyroscope and accelerometer sensor to construct a real position [14]. The actual situation is used as a reference to maintain the TWBR balance. To keep the balance, TWBR will move the motors connected to the wheels. The wheels' movement back and forth requires a control system to maintain stable speed and direction of motor rotation [15]. The control system will be given an additional filter, and the filter functions to reduce noise from sensor readings [16].

This paper presents a TWBR with a PID (proportional integral derivative) control method to keep the robot's body balanced in a perpendicular

position to the earth's surface during movement. This system has two inputs, namely accelerometers, used to measure angular acceleration ( $\text{m/s}^2$ ), and gyroscopes, for measuring angular velocity ( $\text{rad/s}$ ) [17]. The accelerometer and gyroscope values were calculated using the complementary filter method to obtain the angle values. The angle obtained is then compared with the setpoint, which is  $0^\circ$ . The difference between the setpoint and the complementary filter angle is processed using the PID control method [16]. The PID control process results are used to regulate the rotation of the wheel drive motor in the robot. The direction of the wheel drive motor rotation will be forward if the complementary filter angle is less than zero and reverse if it is more than zero.

## 2. RESEARCH METHODOLOGY

The research method used is the study of literature and laboratory experiments. The technique used in this study is shown in Figure 1. This study uses a flowchart where there are sequences of work carried out. The first stage is Pre-research, wherein in this section, available research is carried out on the subject that will be carried out in this research. The second stage is to identify problems with clear boundaries, usually using literature study techniques or field observations. The third stage is to do the analysis, where after all the data is obtained through the identification stage, the next step is to design the hardware and software of the robot. The fourth stage is system design, starting from designing the mechanical structure of the robot, robot software, and android software. The fifth stage is implementation, which transfers the design results at the previous stage into the system. The sixth stage is testing, wherein the hardware and software are tested in this stage.

motor will provide the appropriate strength so that the pendulum remains upright [22].

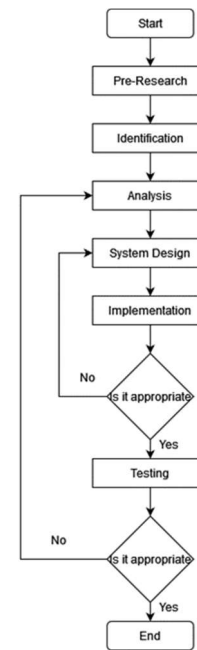


Figure 1: Research Method Flow Chart

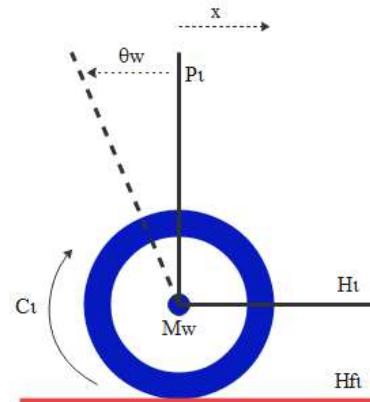


Figure 2: Model of An Inverted Pendulum [23]

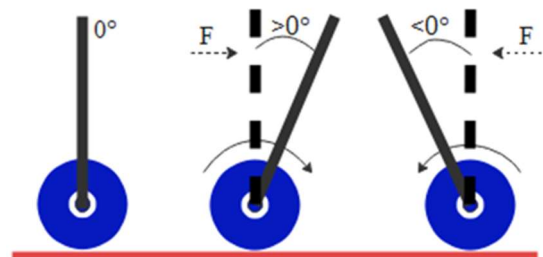


Figure 3: Motor Rotation Direction

Figure 2 is a diagram of the force analysis of an inverted pendulum. The inverse pendulum force

equation can be obtained by Newton's law [24] and the rotational torque formula [25]:

The right wheel force equation is as follows:

$$\begin{aligned} M_w \ddot{x} &= H_{fR} - H_R \\ I_w \ddot{\theta}_w &= C_R - H_{fR} \cdot R \end{aligned} \quad (1)$$

The left wheel force equation is as follows:

$$\begin{aligned} M_w \ddot{x} &= H_{fL} - H_L \\ I_w \ddot{\theta}_w &= C_L - H_{fL} \cdot R \end{aligned} \quad (2)$$

After finishing, the following is obtained:

$$2 \left( M_w + \frac{I_w}{R^2} \right) \ddot{x} = \frac{C_R + C_L}{R} - (H_R + H_L) \quad (3)$$

where  $\ddot{x}$  is the wheel acceleration of x axis;  $M_w$  is the weight of the wheel;  $C_R$  and  $C_L$  are the right and left wheel torque;  $\theta_w$  is the angle of the wheel around the gravity axis direction;  $H_R$  and  $H_L$  are the gravity axis forces of the left and right wheels with the car body;  $I_w$  is the moment of inertia of the wheel; and  $H_{fR}$  and  $H_{fL}$  are the interatomic forces of the right and left wheels with the ground.

An illustration of the robot work system is shown in Figure 3. If the angle reading is smaller than  $0^\circ$ , the motor rotates counterclockwise, whereas for a bigger angle reading than  $0^\circ$ , the motor turns clockwise.

## 2.2 Complementary Filter

Complementary filters are digital filters that can eliminate the effects of signal interference (noise) and drift [26][27] from the readings of the accelerometer and gyroscope [28][29]. The accelerometer can accurately measure the tilt angle when the system is at rest (static) but not accurately when it is in motion (dynamic) due to the influence of earth's gravity and external vibrations. The gyroscope can measure dynamic angular velocity. However, it is not accurate in the long run due to the resulting bias effect.

The working principle of a complementary filter is shown in Figure 4. The signal from the accelerometer measurement in the form of an angle value has noise at high frequencies, so it is filtered with a low pass filter (LPF). The signal from the gyroscope measurement in the form of angular velocity has noise at low frequencies, so it is filtered with a high pass filter (HPF). The sum of the two filtered signals is an angle ( $\theta$ ).

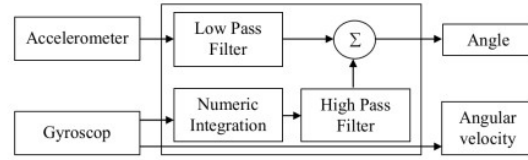


Figure 4: Block Diagram of The Complementary Filter

The equation of the complementary filter is shown in (4):

$$\theta = \alpha * (\theta + gyroAngle * sampleTime) + (1 - \alpha) accelerometerAngle \quad (4)$$

where  $\theta$  is the current angle of the robot and  $\alpha$  is the filter coefficient, which determines to which degree the final angle measurement will depend on each sensor.

## 2.3 Ziegler-Nichols Tuning Second Method

This method is used in the close loop system, which is a PID control tuning method. The PID controller is characterized by three parameters:  $K_p$ ,  $K_i$ , and  $K_d$ , respectively called a proportional constant, an integral constant, and a derivative constant [30].  $K_p$  is a proportional gain constant that will accelerate the response and is always proportional to the drive error signal's magnitude.  $K_i$  is an integral strengthening constant that will improve the steady-state response.  $K_d$  is a constant of derivative gain and giving the value of  $K_d$  can reduce the overshoot caused by the system.

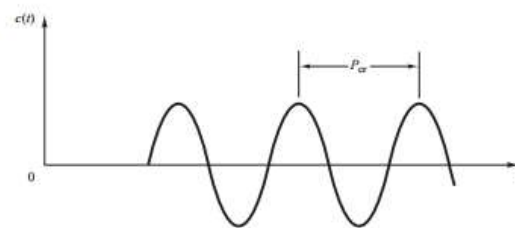


Figure 5: Sustained Oscillation with Period  $P_{cr}$  ( $P_{cr}$  in second)

The Ziegler-Nichols Tuning Second Method is suitable for systems that experience oscillations. The first thing that must be done to determine the PID tuning is to assign 0 to the  $K_i$  and  $K_d$  parameters, in contrast to the  $K_p$  parameter, until a stable oscillation is obtained. The value of  $K_p$  at this stable oscillation is called the critical gain ( $K_{cr}$ ). Then, the value of the oscillation period ( $P_{cr}$ ) between the oscillation wave peaks is determined (Figure 5). After getting the  $K_{cr}$

and  $P_{cr}$  values, the values of  $K_p$ ,  $K_i$ , and  $K_d$  parameters are determined based on Table 1.

Table 1: The Values of  $K_p$ ,  $K_i$ , and  $K_d$  Parameters

Type of Controller	$K_p$	$K_i$	$K_d$
P	$0.5K_{cr}$	$\infty$	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

## 2.4 PID Controller

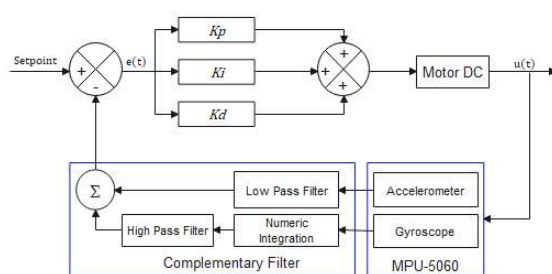


Figure 6: Block Diagram of The PID Control System

Based on Figure 6, systematically, if  $e(t)$  marks an error signal and  $u(t)$  is a control signal, then the PID controller output can be written as follows:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (5)$$

In discrete form, equation (5) can be written as follows:

$$u(k) = K_p e(k) + K_i T_i \sum_{i=0}^k e_i + K_d \frac{e(k) - e(k-1)}{T_d} \quad (6)$$

with  $T_i$  and  $T_d$  denoting the integration and differential times, respectively. Furthermore, Equation (6) is translated into the form of a program code as follows:

$$U = K_p * error + K_i * T_s * (error + lastError) + \frac{K_d}{T_s} * (error - lastError) \quad (7)$$

with  $U$  representing the control signal generated every time; the *error* is the difference between the reference value and the output value for each discrete time; and *lastError* marks the *error* value of one previous counting period. Meanwhile,  $T_s$  states the counting period of the signal processing device in the microcontroller.

## 2.5 Android Navigation System

Figure 7 is a block diagram of a robot navigation system with an Android smartphone. It functions as a robot controller connected wirelessly with the Bluetooth module (HC-05). Robot control takes the form of data communication between an Android smartphone and a microcontroller. Data sent from an Android smartphone is in the form of individual characters with a transmission speed of 9600 bps.

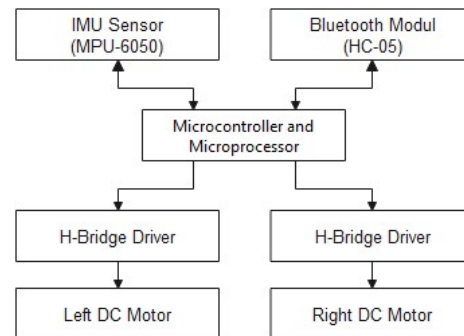






Figure 7: Block Diagram of The Navigation System with The HC-05 Module

Table 2: Command Codes / Characters

Button	Data sent from the smartphone to the microcontroller	The command for the robot
	"w"	Forward
	"s"	Rewind
	"L"	Left
	"R"	Right

The program contained in the Android smartphone is a program that is downloaded from Play Store, known as Arduinio Car (Figure 8). The code/character sent from the Android smartphone to the microcontroller is presented in Table 2. The program starts by opening a Bluetooth connection between the Android smartphone and the HC-05 module. After connecting, four commands to control the robot can be given via the Android smartphone. Each control button on the Android smartphone will send a different character so that the microcontroller can recognize the command given. For example, the

forward base will send the character "w", the reverse order will send the character "s", the command turn left will send the character "L", and the command turn right will send the character "R". The flowchart for controlling the robot movement from an Android smartphone is shown in Figure 9, and the flowchart for controlling the robot movement from the microcontroller is shown in Figure 10.

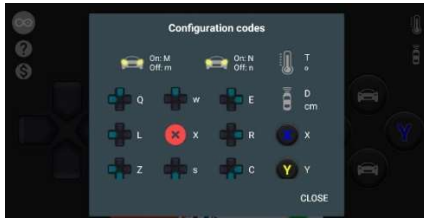


Figure 8: Display GUI in an Android Smartphone

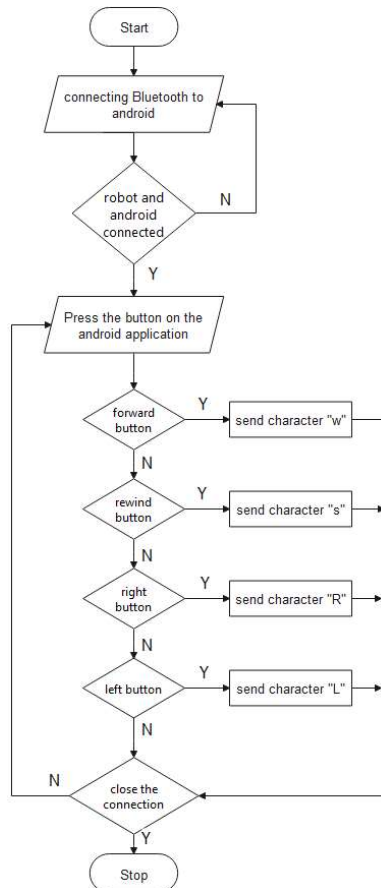


Figure 9: Flowchart of The Robot Motion Control Program from an Android Smartphone

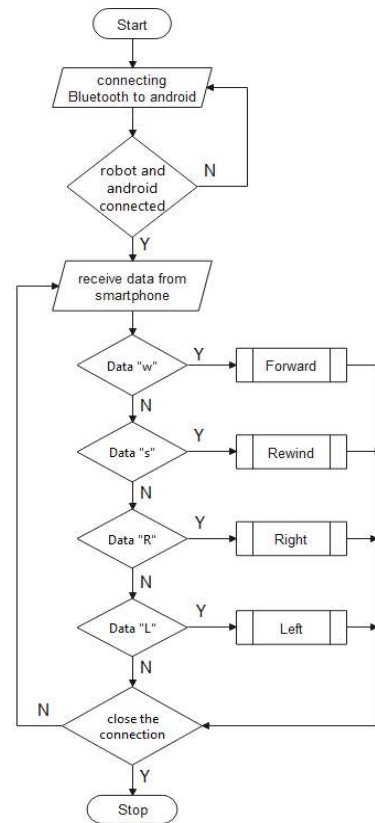


Figure 10: Flowchart Program from The Microcontroller

### 3. EXPERIMENTAL RESULTS

#### 3.1 Two-Wheeled Balancing Robot

The TWBR design is shown in Figure 11(a), and the physical form of TWBR is shown in Figure 11(b). TWBR uses aluminum as a frame or body frame. TWBR also uses acrylic as housing to protect all electrical components. TWBR is 55 cm long, 41 cm high, and 25 cm wide. The total weight of the TWBR is 22 Kg.

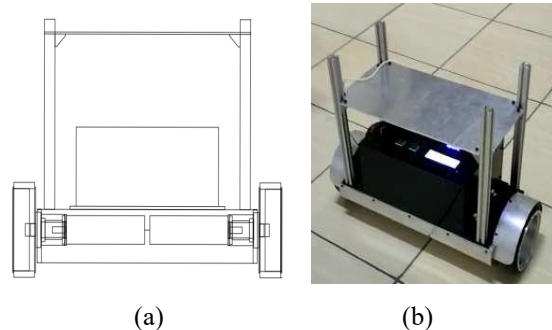


Figure 11: TWBR Design (a), and The Physical Form of The TWBR (b)



The power source is the Lifepo4 6S battery, and the battery has a voltage of 18 Volt DC and a current of 2.5 Ampere. The battery is the most massive component. Therefore, it is placed at the bottom of the robot. This placement aims to make the robot's center of gravity lower to help the robot maintain its balance correctly.

### 3.2 Complementary Filter Testing

The complementary filter test is carried out to see the effect of changing the filter coefficient of the issued angle value. The value of the filter coefficient varies, including  $\alpha = 0.95$ ,  $\alpha = 0.96$ , and  $\alpha = 0.97$ . Based on Figure 12, when  $\alpha = 0.95$ , the filter output has more noise. If  $\alpha = 0.97$ , then the filter output will exceed the sensor's reading; this is shown in Figure 14. Therefore, in this study, the filter coefficient value used is  $\alpha = 0.96$  (Figure 13).

### 3.3 Testing of Ziegler-Nichols Tuning 2<sup>nd</sup> Method

TWBR has a dynamic angle that must always be controlled in order to achieve balance. In this

system, the robot rotates about the Y-axis, so the accelerometer requires two axes, namely X and Z. Meanwhile, the gyroscope takes one axis, namely the Y-axis. The angle value of the complementary filter algorithm is compared with the setpoint value. The difference in value is the error value used in the PID algorithm. The greater the error value, the greater the PID algorithm's result, which can lead to the generation of a more excellent PWM value. This is because a more significant angle of deviation needs a more excellent PWM value to return the pendulum to an upright position.

When P control is used, a large  $K_p$  (proportional gain) must correct the steady-state error. If the  $K_p$  value continues to increase, the oscillation becomes more significant, and the system will become unstable. However, if the  $K_p$  is too small, the system response to the changes will be slow. The steps for tuning the PID constants using the Ziegler-Nichols 2 oscillation method provide variations in the increase of the  $K_p$  value ranging from 0.5, 1, 1.5, 2, 2.5, 3, 3.5, to 4. When the system has experienced a stable oscillation, the  $K_p$  value is stored as a  $K_u$  value.

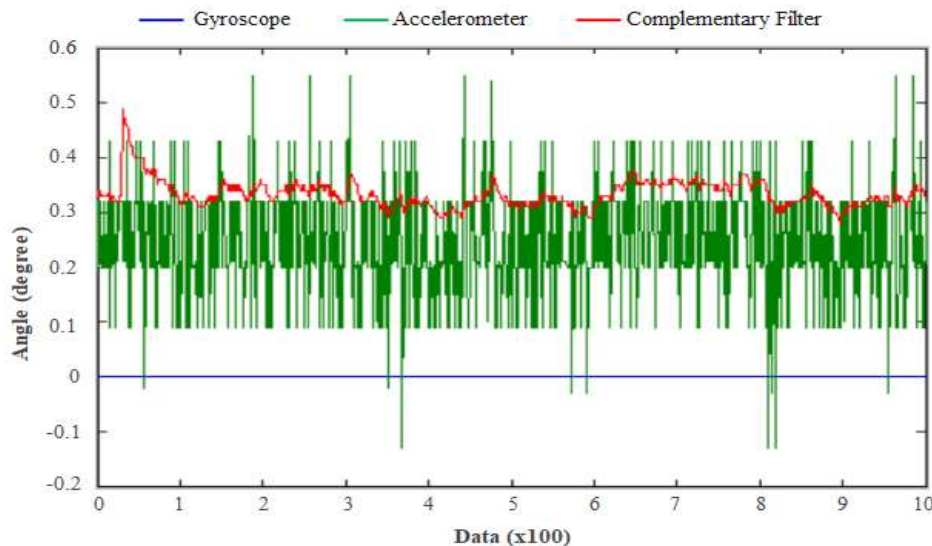


Figure 12: Testing The Complementary Filter Algorithm at An Angle of  $0^\circ$  with a Coefficient of  $\alpha=0.95$

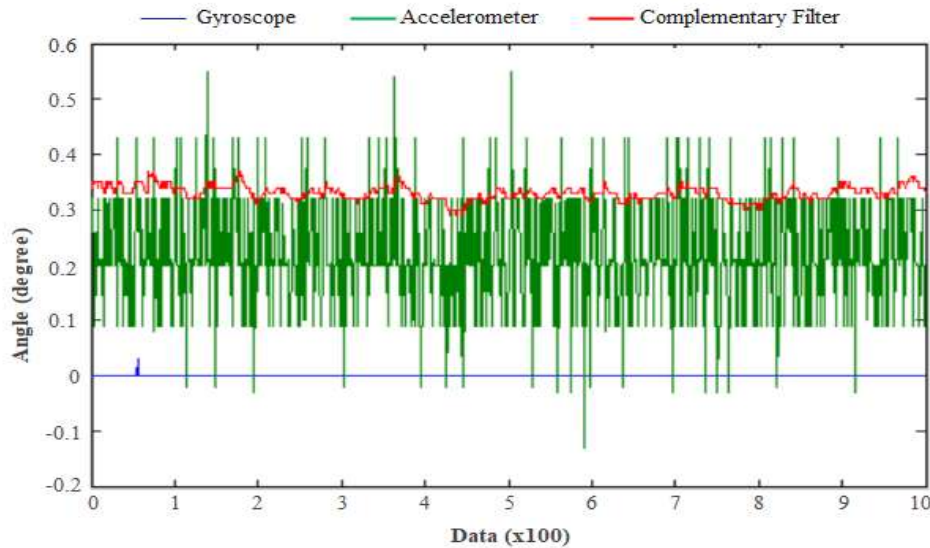


Figure 13: Testing The Complementary Filter Algorithm at An Angle of  $0^\circ$  with a Coefficient of  $\alpha=0.96$

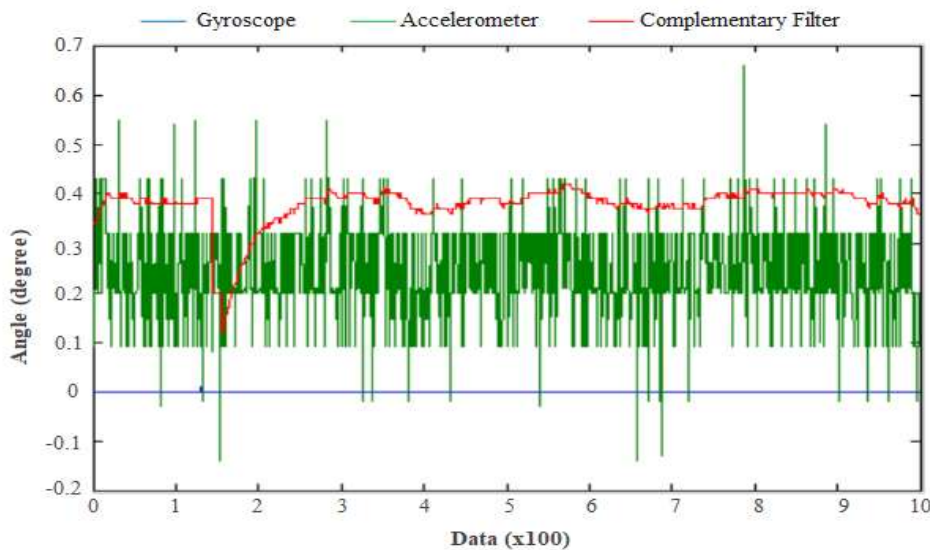


Figure 14: Testing The Complementary Filter Algorithm at An Angle of  $0^\circ$  with a Coefficient of  $\alpha=0.97$

Based on the graph in Figure 15, the oscillation that occurs is towards infinity and is still unstable. Based on the graph in Figure 16, the resulting oscillations are getting bigger and bigger, except for the graph with a value of  $K_p = 2.5$ , which has a reasonably stable oscillation.

The proportional constant in the system is to increase the overshoot value (a condition that exceeds the limit or spike value). The greater the proportional constant entered into the system, the greater the overshoot value, and the system becomes

unstable (far from the setpoint). The constant in question is a derivative constant so that the proportional constant cannot be used alone or balanced with other constants as its damping.

After knowing  $K_p$ 's value, the calculation of the values of  $K_p$ ,  $K_i$ , and  $K_d$  is based on the Ziegler-Nichols 2<sup>nd</sup> tuning method in Table 1. The value of  $K_p = 2.5$ , based on Figure 15, is stored as  $K_{cr}$ . Therefore, the value of  $K_p = 1.5$  and  $P_{cr} = 0.4$  seconds; thus, based on the calculations in Table 3,

we get the value of  $K_i = 0.2$  and the value of  $K_d = 0.05$ .

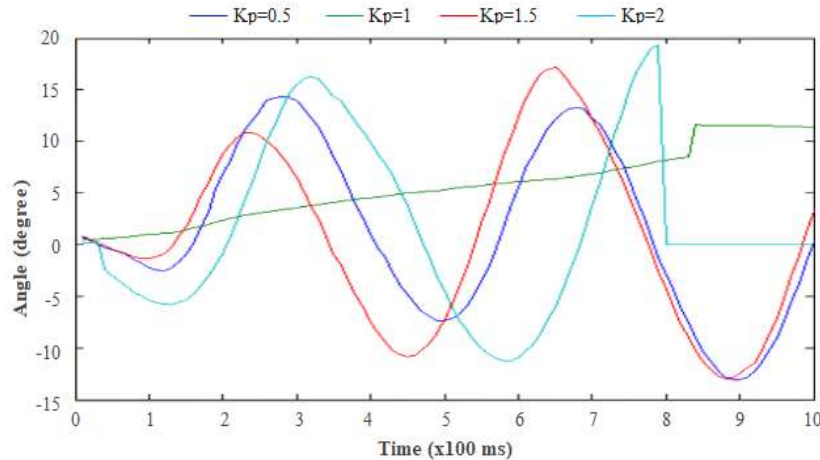


Figure 15: Graph of System Oscillation with  $K_p=0.5$  to  $K_p=2$

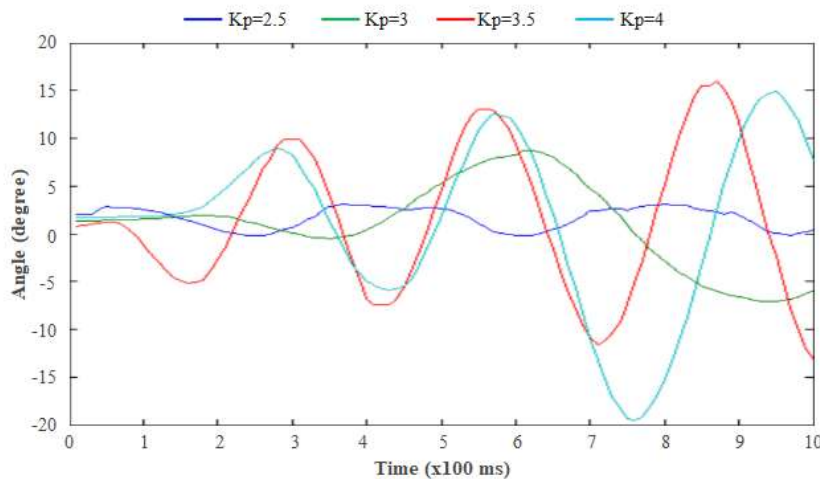


Figure 16: Graph of System Oscillation with  $K_p=2.5$  to  $K_p=4$

Table 3: The Result of The PID Constant

Type of Controller	$K_p$	$K_i$	$K_d$
PID	0.6 (2.5)	0.5 (0.4)	0.125 (0.4)
Result	1.5	0.2	0.05

### 3.4 PID Control Testing

PID control testing is carried out by providing the  $K_p$ ,  $K_i$ , and  $K_d$  values obtained based on the Ziegler-Nichols 2<sup>nd</sup> tuning method in Table 3. Based on the graph in Figure 17, the system initially experiences an oscillation and stabilizes (settling

time) at 10x100ms or 1 second. The overshoot produced by the system tends to be smaller because the  $K_d$  value serves to reduce overshoot. The resulting steady-state error is relatively small, less than 1°, while the setpoint is set to a value of 0°.

To see the control response further, load variation above the robot is applied. Variations in the load are 30 Kg, 40 Kg, and 50 Kg. Table 4 shows the differences in response that occur with the given load variations. The maximum loading that can be given to the robot is 40 Kg. A load of 40 Kg shows a settling time of 1.21 seconds with a steady-state error of about 2.06%. The interpretation is that at the



load of 40 Kg, the robot's movement is still stable to maintain its balance with a maximum error of  $1.5^\circ$ .

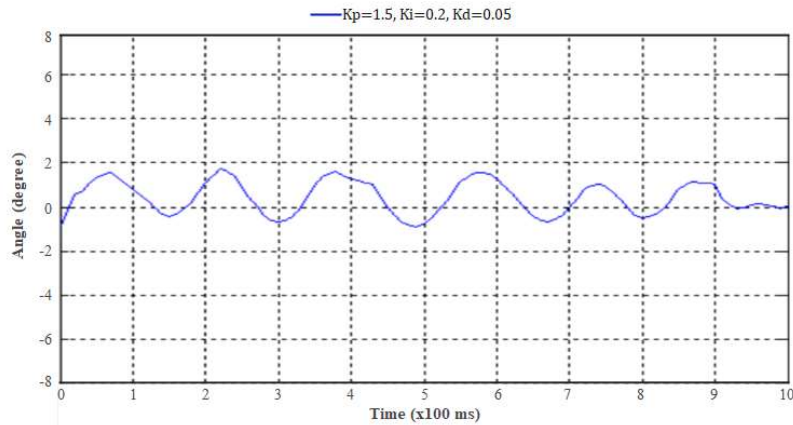


Figure 17: Graph of PID Response

Table 4: System Response with Load and Non-Load Variations

Response Specifications	Load (Kilogram)			
	0	30	40	50
Rise Time	6.835 ms	4.047 ms	4.382 ms	5.364 ms
Peak Time	0.04 s	0.04 s	0.05 s	0.06 s
Settling Time	0.79 s	1.09 s	1.21 s	2.42 s
Steady State Error	1.60 %	1.86 %	2.07 %	2.76 %

When given a load of 50 kg, the maximum value of overshoot is the largest among other load variations, namely 14.64%. This shows that there is an increase in overshoot as the burden on the robot increases. At a load of 50 kg, the settling time is around 2.42 seconds, with a steady-state error of 2.76%. It shows that the robot tends to show the smallest mistake and signal fluctuation at this loading and reaching a steady-state requires a longer time than other load conditions.

#### 4. DISCUSSION

This paper proposes a TWBR that can be controlled via Android Smartphone. The principles and technology in making TWBR proposed in this paper are intended for future research. In addition, the proposed TWBR system can be applied in

various fields by adding some additional instruments or sensors, such as camera sensors.

Based on the research results, TWBR will stand in a balanced position when getting a load between 0-40 Kg. The balanced position is obtained by adequately tuning the PID. So that when the robot tends to the left, the DC motor provides torque counterclockwise, returning the robot to its upright position. Conversely, when the robot tends to the right from its proper place, the DC motor provides torque clockwise to maintain its balance position. The mechanism for testing the robot in a balanced position is continuously observed, and observations are made when the TWBR is in a stationary or moving part, and there is no load on it. Due to the non-linear nature of the inverted pendulum system, the non-linear control approach is suitable for this TWBR control technique. As consideration for further research, the application of PID combination control techniques with intelligent control techniques with neural networks as described in [12] can be used as material for further research to obtain better TWBR performance. In addition to further research, the PID control constant tuning method is carried out with the auto-tuning method so that the results obtained are more optimal than in this study. Furthermore, the fusion sensor method uses other algorithms such as the Kalman Filter or DCM (Discrete Cosine Matrix) compared with the complementary filter method.

## 5. CONCLUSIONS

Based on the research done, it can be concluded that TWBR can control balance using PID control with  $K_p = 1.5$ ,  $K_i = 0.2$ , and  $K_d = 0.05$ . TWBR can be balanced with no-load or under load, stationary, and moving. No-load condition is a stable response with a settling time of 0.79 seconds. The steady-state error is 1.6%. However, when given a 40 kg load, the robot shows irregular movements and tends to fall. The robot's movements can be appropriately controlled via an Android smartphone. The robot can move in a balanced manner and is able to withstand an angle range of  $-1.5^\circ$  to  $1.5^\circ$ , provided that the direction of the robot or the rotation speed of its wheel is constant.

## 6. ACKNOWLEDGEMENTS

The author would like to thank those who have contributed/supported this research, namely the Institute for Research and Community Service (LPPM), State University of Medan, where DIPA Unimed funded this research in the 2020 fiscal year.

## REFERENCES:

- [1] F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer, "JOE: A mobile, inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 107–114, 2002, doi: 10.1109/41.982254.
- [2] F. Kung, "A Tutorial on Modelling and Control of Two- Wheeled Self-Balancing Robot with Stepper Motor," vol. 3, no. 2, pp. 64–73, 2019.
- [3] I. Jmel, H. Dimassi, S. Hadj Said, and F. M'Sahli, "Adaptive Observer-Based Output Feedback Control for Two-Wheeled Self-Balancing Robot," *Mathematical Problems in Engineering*, vol. 2020, 2020, doi: 10.1155/2020/5162172.
- [4] A. S. Wardoyo, S. Hendi, D. Sebayang, I. Hidayat, and A. Adriansyah, "An investigation on the application of fuzzy and PID algorithm in the two wheeled robot with self balancing system using microcontroller," *Proceedings - 2015 International Conference on Control, Automation and Robotics, ICCAR 2015*, pp. 64–68, 2015, doi: 10.1109/ICCAR.2015.7166003.
- [5] J. Fang, "The LQR controller design of two-wheeled self-balancing robot based on the particle swarm optimization algorithm," *Mathematical Problems in Engineering*, vol. 2014, 2014, doi: 10.1155/2014/729095.
- [6] R. P. M. Chan, K. A. Stol, and C. R. Halkyard, "Review of modelling and control of two-wheeled robots," *Annual Reviews in Control*, vol. 37, no. 1, pp. 89–103, 2013, doi: 10.1016/j.arcontrol.2013.03.004.
- [7] J. Han, X. Li, and Q. Qin, "Design of two-wheeled self-balancing robot based on sensor fusion algorithm," *International Journal of Automation Technology*, vol. 8, no. 2, pp. 216–221, 2014, doi: 10.20965/ijat.2014.p0216.
- [8] S. Miasa, M. Al-Mjali, A. A. H. Ibrahim, and T. A. Tutunji, "Fuzzy control of a two-wheel balancing robot using DSPIC," *2010 7th International Multi-Conference on Systems, Signals and Devices, SSD-10*, pp. 7–12, 2010, doi: 10.1109/SSD.2010.5585525.
- [9] J. Wu and W. Zhang, "Design of fuzzy logic controller for two-wheeled self-balancing robot," *Proceedings of the 6th International Forum on Strategic Technology, IFOST 2011*, vol. 2, pp. 1266–1270, 2011, doi: 10.1109/IFOST.2011.6021250.
- [10] C. H. Chiu, Y. W. Lin, and C. H. Lin, "Real-time control of a wheeled inverted pendulum based on an intelligent model free controller," *Mechatronics*, vol. 21, no. 3, pp. 523–533, 2011, doi: 10.1016/j.mechatronics.2011.01.010.
- [11] S. W. Nawawi, M. N. Ahmad, and J. H. S. Osman, "Real-Time Control System for a Two-Wheeled Inverted Pendulum Mobile Robot," in *Proceedings of World Academy of Science, Engineering and Technology*, 2008, vol. 29, no. May, pp. 214–220, doi: 10.5772/10362.
- [12] A. Unluturk and O. Aydogdu, "Adaptive control of two-wheeled mobile balance robot capable to adapt different surfaces using a novel artificial neural network-based real-time switching dynamic controller," *International Journal of Advanced Robotic Systems*, vol. 14, no. 2, pp. 1–9, 2017, doi: 10.1177/1729881417700893.
- [13] K. Prakash and K. Thomas, "Study of controllers for a two wheeled self-balancing robot," *2016 International Conference on Next Generation Intelligent Systems, ICNGIS 2016*, 2017, doi: 10.1109/ICNGIS.2016.7854009.
- [14] J. H. Park and B. K. Cho, "Development of a self-balancing robot with a control moment gyroscope," *International Journal of*

- Advanced Robotic Systems*, vol. 15, no. 2, pp. 1–11, 2018, doi: 10.1177/1729881418770865.
- [15] M. T. A. Rahman and S. Ahmad, "Performance comparison between PD-fuzzy and PID controller towards the stability of the extendable double-link two-wheeled mobile robot," *2015 10th Asian Control Conference: Emerging Control Techniques for a Sustainable World, ASCC 2015*, 2015, doi: 10.1109/ASCC.2015.7244892.
- [16] J. Zhang, S. Wei, B. Zhou, and Y. Song, "Study on sliding mode trajectory tracking control of mobile robot based on the kalman filter," *2016 IEEE International Conference on Information and Automation, IEEE ICIA 2016*, no. August, pp. 1195–1199, 2017, doi: 10.1109/ICInfA.2016.7832001.
- [17] H. S. Juang and K. Y. Lurr, "Design and control of a two-wheel self-balancing robot using the arduino microcontroller board," *IEEE International Conference on Control and Automation, ICCA*, no. June, pp. 634–639, 2013, doi: 10.1109/ICCA.2013.6565146.
- [18] A. M. Almeshal, K. M. Goher, and M. O. Tokhi, "Dynamic modelling and stabilization of a new configuration of two-wheeled machines," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 443–472, 2013, doi: 10.1016/j.robot.2013.01.006.
- [19] A. N. Inal, Ö. Morgül, and U. Saranlı, "Path following with an underactuated self-balancing spherical-wheel mobile robot," *Proceedings of the 17th International Conference on Advanced Robotics, ICAR 2015*, pp. 194–199, 2015, doi: 10.1109/ICAR.2015.7251455.
- [20] D. Choi and J. H. Oh, "Four and two wheel transformable dynamic mobile platform," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 10–13, 2011, doi: 10.1109/ICRA.2011.5980580.
- [21] K. Pathak, J. Franch, and S. K. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 505–513, 2005, doi: 10.1109/TRO.2004.840905.
- [22] Z. Li, "Adaptive fuzzy output feedback motion/force control for wheeled inverted pendulums," *IET Control Theory and Applications*, vol. 5, no. 10, pp. 1176–1188, 2011, doi: 10.1049/iet-cta.2010.0176.
- [23] J. Dong, B. He, C. Zhang, and G. Li, "Open-Closed-Loop PD Iterative Learning Control with a Variable Forgetting Factor for a Two-Wheeled Self-Balancing Mobile Robot," *Complexity*, vol. 2019, no. Ilc, 2019, doi: 10.1155/2019/5705126.
- [24] A. M. Bloch and P. E. Crouch, "Newton's law and nonholonomic systems," *Proceedings of the IEEE Conference on Decision and Control*, vol. 4, no. December, pp. 3569–3574, 1998, doi: 10.1109/cdc.1998.761733.
- [25] E. R. Wuori and J. H. Judy, *Rotational Hysteresis for D O M a I N Wall Motion I N the*, vol. M, no. 5. 1985.
- [26] W.-K. Chen, *The Electrical Engineering Handbook*, 1st ed. UK: Elsevier Academic Press, 2004.
- [27] A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*, 2nd ed. Academic Press Elsevier, 2001.
- [28] A. Noordin, M. A. M. Basri, and Z. Mohamed, "Sensor fusion algorithm by complementary filter for attitude estimation of quadrotor with low-cost IMU," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 2, pp. 868–875, 2018, doi: 10.12928/TELKOMNIKA.v16i2.9020.
- [29] L. Yang, J. Ren, K. Song, X. Ma, and X. Yang, "Complementary Filter Design Based Error Allocation for Attitude Estimation with Low-Cost Sensor," vol. 140, no. Ecae 2017, pp. 206–211, 2018, doi: 10.2991/ecae-17.2018.43.
- [30] S. Khatoon, D. K. Chaturvedi, N. Hasan, and M. Istiyaque, "Optimal controller design for two wheel mobile robot," *3rd International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity, CIPECH 2018*, pp. 155–159, 2018, doi: 10.1109/CIPECH.2018.8724314.