STUDY THE PERFORMANCE OF TWO-WHEELED BALANCING MOBILE ROBOT USING FUZZY PD CONTROLLER

Nurul Muthmainnah Mohd Noor*, Saiful Amirul Asyraf bin Mohd Halid, and Rohidatun Mahmod @Wahab

Mechanical Engineering Studies, College of Engineering, Universiti Teknologi MARA, Penang Branch, Permatang Pauh Campus, 13500, Permatang Pauh, Penang, Malaysia

*Corresponding email: muthmainnah@uitm.edu.my

Article history
Received
1st May 2023
Revised
29th October 2023
Accepted
23rd November 2023
Published
1st December 2023

ABSTRACT

Nowadays, the research on a two-wheeled self-balancing robot is an active area of research especially in terms of design as well as control to continue the innovation applications of robots in the future. Most of the two-wheeled self-balancing robots are designed based on an inverted pendulum system for stability and maneuverability. The aim of this paper is to propose the fuzzy PD controller to control and maintain its balance on the two wheels. A sensor of the Inertial Measurement Unit (IMU) was used as an input to evaluate and obtain the position and orientation of the robot. The control algorithms for the robot also are designed to keep the pendulum upright. Then, the fuzzy PD concept was applied to correct the error between the desired set point and the actual tilt angle position to adjust the speed of the motor accordingly. The results obtained from this controller were capable of maintaining the balancing of the robot by using an experimental method of PID tuning. The prototype of the two-wheeled self-balancing robot was implemented with Arduino Uno and a fuzzy PD controller. However, the limitation of the project is the longer size and heavier weight of the robot are less stable, then a better controller is needed to balance the robot.

Keywords: *Fuzzy PD controller, two-wheeled self-balancing, mobile robot, performance*

1.0 INTRODUCTION

Working on two-wheeled self-balancing robots has increased in recent years due to the invention of Segway, which is a human carrier device [1-2]. The development of the two-wheeled self-balancing robot uses sensors such as accelerometers, gyroscopes, and encoders, meanwhile, motors act as an actuator to maintain the upright position. The combination of sensors and motors can control the movement of the robot by adjusting the robot's speed and direction [3]. The criteria for choosing the sensor is based on the purpose of the project, for example, the accelerometers measure the robot's acceleration. By using the gyroscopes, it will measure and maintain the angular velocity and orientation of the object [4].

The MPU-650 is an IMU sensor that is widely used in the development of two-wheeled mobile robots. This sensor has a built-in accelerometer, gyroscope, and sometimes magnetometer. The sensor is capable of accurately and precisely detecting signals, even when it is relatively small and sensitive to noise [5]. This device uses an accelerometer to measure a linear acceleration in a particular direction. In the meantime, the gyroscope is also utilized for calculating the angular velocity of rotation around the axis. In order to determine the orientation of the device relative to

 DOI: https://doi.org/10.11113/jm.v46.500

the Earth's magnetic field, the magnetometer will fit together in the IMU sensor. The concept of the IMU sensor is based on the principle of inertia. When an object is accelerating, the acceleration of the device can be determined by the inertia force, which is directly proportional to the acceleration. Similarly, when an object is in rotation, the gyroscopic force will come into play to ascertain the angular velocity of the device [6-7]. Therefore, the IMU sensor can be applied for any application in navigation, virtual reality, and sports [8].

In the previous study, numerous concepts are available and can be put into application for controlling the two-wheeled self-balancing robot either using PID itself or a combination of fuzzy PD controllers [9-11]. In this study [12], the findings said that the fuzzy-PID controller has a better result in the nonlinear properties of the wind speed and pitch angle. This article mentions the PID controller was used to control the angle by implementing a fuzzy logic controller to tune their parameters. The nonlinear properties of the wind speed and pitch angle can be controlled and analysed by using a PID-fuzzy controller. The other method, both PID and LQR control techniques also can be employed to manage the position and velocity that enable the robot to maintain vertical balance exclusively [13]. The PID controller is easier to implement but the main concern is the overshoot performance makes the system prone to disturbance.

In the paper of T.A.Mai, et al [14], they discussed the combination of PID and fuzzy logic control as a powerful tool for designing the nonlinear system of the two-wheeled balancing robot. The fuzzy controller was designed based on the relation models to control the robot's balancing as well as the robot's position by implementing the STM32F4 Discovery Kit for real-time operation. Another research has been done by Khan, et al [15], they said that the PID controllers were designed using Social Spider Optimization (SSO) algorithm to demonstrate the effectiveness of these controllers in order to optimise the speed of the motor of the robot [15].

One of the challenges in the study of two-wheeled self-balancing robots is designing a simple robot that can be both steady and controllable [16-17]. Therefore, reducing the size and weight of the robot also needs to be considered in maintaining its performance, especially in speed. In terms of controlling, the algorithms that have been made must be robust to disturbances and expected inputs like wind, the surface of the path, and the surroundings. Therefore, the purpose of the fuzzy PID controller was introduced to control and balance the robot during movement in turning right and left direction. The two-wheeled self-balancing robot was designed in SOLIDWORK and then fabricated to be a prototype. This prototype was implemented with the fuzzy PD controller to control the speed of movement and tilt angle for balancing the robot. The Arduino was used as an interface between the prototype (mechanical part) and the IMU sensor (electrical part).

2.0 METHODOLOGY

2.1 Conceptual Design based on Inverted Pendulum

The principle of the inverted pendulum was introduced in developing the control algorithm for two-wheeled self-balancing robots. The concept of an inverted pendulum is a physical system that is unstable but can be balanced by applying forces at the appropriate times [18]. Figure 1 shows the model of a simple inverted pendulum. The model consists of a mass which is a robot's body suspended from a pivot. The mass is free to rotate about the pivot and the system is subject to gravity [19 - 20].

DOI: https://doi.org/10.11113/jm.v46.500

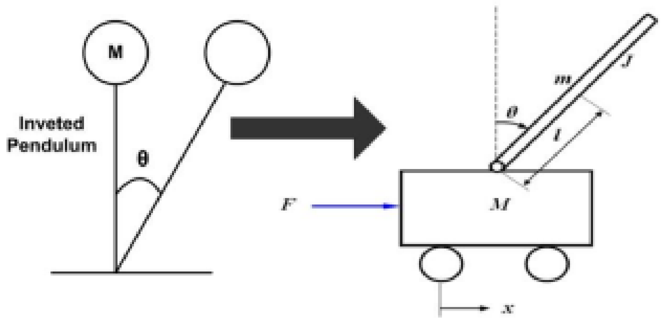


Figure 1: The model of a simple pendulum [18].

The inverted pendulum system can be used to analyse the stability of the robot by using the equation of (1).

$$m\ddot{\theta} + b\dot{\theta} + mgl\sin\theta = u \quad (1)$$

where:

m – the mass of the robot
b – damper coefficient
g – the acceleration due to gravity
l – the length of the pendulum
θ – the angle of the pendulum from the vertical
u – the control input (input force)

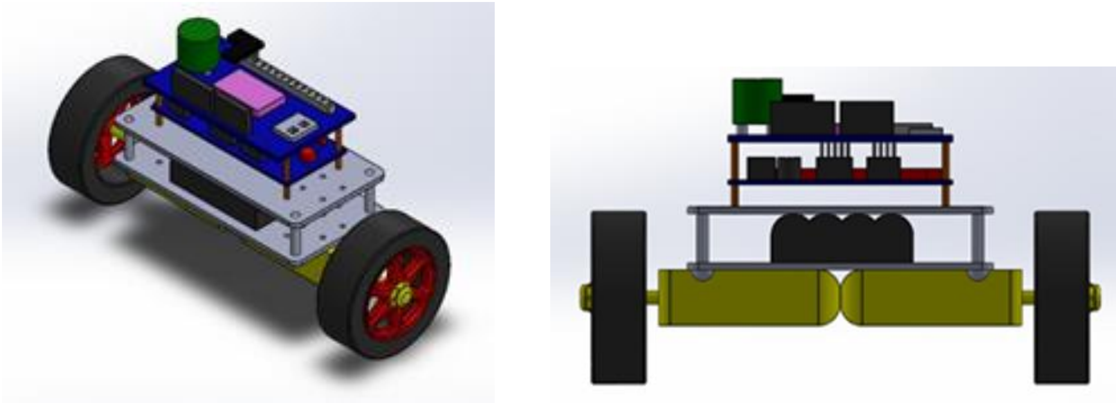
Equation (1) represents the inertia of the pendulum, damping force, which is a force that opposes the motion of the pendulum as well as the gravitational force. The gravitational force will pull the pendulum down and the control input is the force applied to the cart. The main goal of the control system is to apply the force to the cart such that the pendulum always remains upright. Therefore, the angle of the pendulum should be zero.

For the inverted pendulum concept, the stability of the robot can be determined by the length of the pendulum and the mass of the robot. A longer pendulum is more stable compared to a heavier robot [21-22]. Therefore, the control algorithms are designed to keep the pendulum upright. Meanwhile, the output of the sensor will determine the position and orientation of the robot.

2.2 Design a Two-Wheeled Self-Balancing Mobile Robot

In this work, the two-wheeled self-balancing robot was designed using SOLIDWORK as shown in Figure 2. The design was considered to be the possible weight that was distributed on the frame, and it is directly affected by the balancing performance. Based on the design, the robot consists of two plates in a rectangular shape and connected with the two tires. The electronic components such as Arduino Uno, motor driver, and voltage regular were placed in the top plate. Meanwhile, the second plate contains wheels, motors, angle sensors, and a battery. The size of the robot was designed to make sure the robot can independently upright position. Table 1 shows the parameters of the two-wheeled self-balancing robot.

DOI: https://doi.org/10.11113/jm.v46.500



(a) (b)

Figure 2: The model of a two-wheeled self-balancing robot in SOLIDWORK. (a) 3-D view, (b) Front view

Table 1: Parameters of two-wheeled self-balancing robot

Symbol	Parameter	Value
m_m	Motor mass	0.275 kg
K_b	Back EMF constant	0.0458 Vs/rad
K_T	Torque constant	0.0458 Nm/A
K_m	Motor constant	2.90×10^{-2} Nm/W^{-2}
R_T	Resistance	10 Ω
T_F	Fraction torque	0.0056 Nm
I_r	Motor rotor inertia	$7\text{-}10 \times 10^{6}$ $kg.m^2$
Iw	Robot wheel inertia = $(Mw \times R^2)/2$	2.3250
I_L	Robot chassis inertia = $(h^2+d^2)/12$	0.1124
D	Distance between contact of wheels	12 cm
G	Gravitational force	9.81 m/s^2
L	Distance between center of gravity and center of wheels	6 cm
m_P	Chassis mass	0.55 kg
r_w	Wheels radius	3.5 cm

Figure 3 shows the prototype of the two-wheeled self-balancing system that contains the IMU sensor and two motors. In this project, the sensor of MPU 6050 was used. This sensor has a 6 DOF consisting of three accelerometer values and three gyroscope values. For driving the motors, the L298 dual h-bridge driver was used to drive the motor in high voltage and current. Most of this type of H-bridge is able to operate in high working voltage which is it can reach up to 46 V.

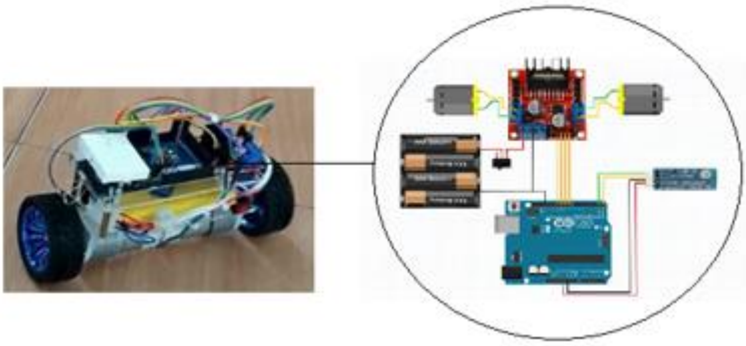


Figure 3: The prototype of a robot with the connection of H-bridge and DC motors

DOI: https://doi.org/10.11113/jm.v46.500

3.3 Mathematical Modelling for Controlling the Robot

In the robotic system, there are two parts that will be studied: electrical and mechanical parameters such as the radius of the wheel, inertia matrix as well as the resistance of the motor. The calculations can made by SOLIDWORK and also by experimental tuned based on the measurement results.

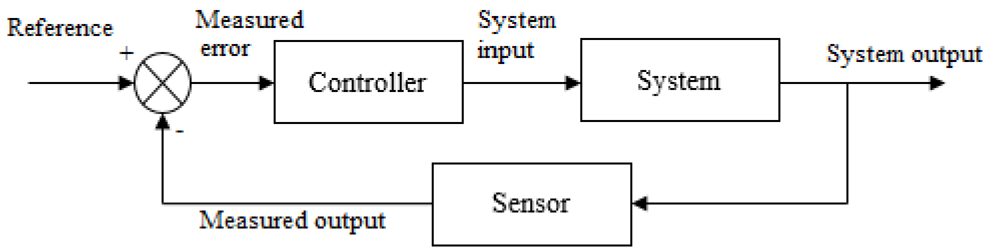


Figure 4: The two-wheeled self-balancing robot system

Figure 4 shows the block diagram for the system. The transfer function for the balancing robot is a mathematical equation that describes the robot's position and tilt angle. This transfer function also can be used to design a controller that will keep the robot balanced and can be written as: -

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_t/R_T I_T}{s^2+s[\frac{K_m}{I_T}+\frac{K_t K_b}{R_T}]} \quad \text{(2)}$$

where K_t, R_T, and I_T are torque constant, resistance, and motor inertia.

Using the parameter in Table 1, the transfer function for this system can be derived as: -

$$\frac{\theta_m(s)}{E_a(s)} = \frac{6.54\ x\ 10^{-4}}{s^2+2.10\ x 10^{-4} s} \quad \text{(3)}$$

This transfer function can be used with θ_1 and θ_2 of the angular displacements for both wheels.

2.4 Fuzzy PD Controller

A fuzzy PD controller is a type of controller that uses fuzzy logic to control the balance and maintenance of a two-wheeled self-balancing robot. It is a type of artificial intelligence that allows the controller to make decisions based on a set of rules. The rules are based on the robot's position and tilt angle [23]. The controller then uses these rules to calculate the amount of speed that needs to be applied to the wheels in order to keep the robot balanced. In the advantages, the fuzzy PD controller has several advantages over traditional PID controllers [24-25]. First, it is more robust to noise and disturbances. Second, it is more adaptable to changes in the robot's environment. Third, it is more efficient in terms of power consumption. The comparison of the PID controller and fuzzy PD controller is shown in Table 2.

Table 2: The comparison between PID controller and Fuzzy PD controller

Criteria	PID Controller	Fuzzy PD Controller
Accuracy	High	High
Stability	High	High
Robustness	Low	High
Adaptability	Low	High
Complexity	Low	High
Cost	Low	High

DOI: https://doi.org/10.11113/jm.v46.500

The steps involved in designing a fuzzy PD controller for the two-wheeled self-balancing robot are shown in Figure 5. In this work, there are two inputs which are position and tilt angle. The speed is controlled by the amount of position and tilt angle that needs to be applied to the wheels. After that, membership functions of the fuzzy rules based on the robot's position and tilt angle are developed. The rules are typically expressed in the form of "if-then" statements. Fuzzification is the process of converting the input variables into fuzzy sets. A fuzzy set is a set of values that have a degree of membership. Defuzzification is the process of converting the output variable from a fuzzy set into a crisp value. The most common defuzzification method is the center of gravity method. Lastly, the fuzzy PD controller can be implemented using an ATmega2560 microcontroller [26-27].

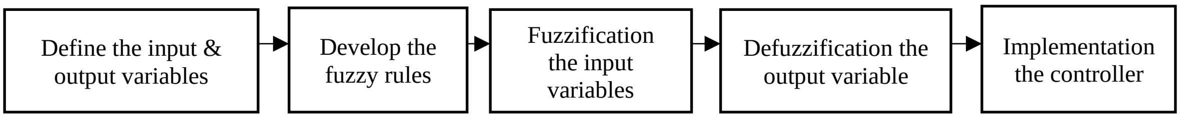


Figure 5: The step in designing a fuzzy PD controller

The fuzzy PD controller is a powerful tool that can be used to control the balance of the mobile robot. It is more robust to noise and disturbances compared to the traditional PID controllers. It is also more adaptable to changes in the robot's environment. The fuzzy rules based on this system as shown in Table 3. The triangular fuzzy membership functions have been used for linguistic terms of 'Low', 'Medium', and 'High'. The simple pattern of adjusting the speed based on both tilt angle and position values.

Table 3: The fuzzy rules based for the system

Tilt angle	Position	Speed
	Low	Low
Low	Medium	Medium
	High	High
	Low	Medium
Medium	Medium	Medium
	High	Medium
	Low	High
High	Medium	High
	High	High

2.5 Experiment Setup of Two-Wheeled Self-Balancing Mobile Robot

In this work, there are two inputs that have been defined: - position and tilt angle. These inputs are typically measured from the sensor of IMU, then the output is speed from the DC motor that controls the signal and adjusts the speed of the system. By using PD-fuzzy based rules, the terms of speed can be considered "low", "medium", and "high". For example, IF the tilt angle is low AND the position is low, THEN the speed should be low.

For proportional gain, K_p, is the difference between the desired change and the current speed, meanwhile, in derivative gain, K_d is the rate of change of desired change. The adjustment of the membership functions, fuzzy rules, and PD control parameters to achieve the desired system performance. The PD controller calculates the motor signal based on the difference between the desired change and the current state of the system. It means that this signal determines how much the motor should adjust its speed to maintain and balance the robot. Figure 6 shows the general block diagram for this system.

DOI: https://doi.org/10.11113/jm.v46.500

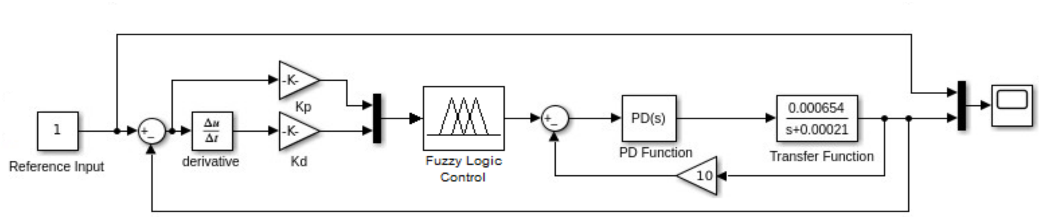


Figure 6: PID-fuzzy controller proposed for the system

When creating fuzzy membership functions, the tilt angle's scope is characterized using three linguistic terms: "low" ranging from -10° to -5°, "medium" covering from -5° to 5°, and "high" encompassing 5° to 10°. Concerning position, the intervals are designated as follows: "low" spanning from -1 to -0.5, "medium" ranging from -0.5 to 0.5, and "high" covering 0.5° to 1. Similarly, for speed control, the divisions are defined as "low" spanning from -100 to -75, "medium" ranging from -75 to 50, and "high" encompassing 50 to 100. After developing the fuzzy rules, this file is converted to Arduino coding. This fuzzy algorithm can be implemented into hardware to control and balance the mobile robot. The turning points of K_p and K_d are tried, and errors are based on the performance of the speed of the motor.

3.0 RESULTS AND DISCUSSION

In general, the two-wheeled self-balancing robot was tested based on the algorithm instruction by utilizing the concept of IoT. Figure 7 shows the robot that was successfully controlled based on GUI. This GUI was developed through the smartphone's Android application.

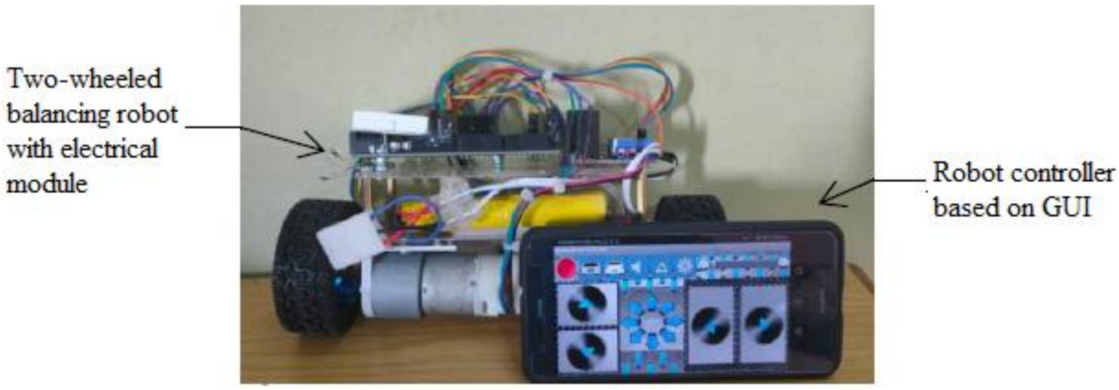


Figure 7: The two-wheeled self-balancing mobile robot and its controller

In this study, the performance of the two-wheeled self-balancing robot was tested by applying a fuzzy PD control method. The fuzzy PD controller can be used for analysing the system which has a high-order, multiple-variable, nonlinear, strong-coupling, and unstable system. The performance of the system can be seen through the performance on settling time, maximum overshoot, and steady state error by changing the parameters. However, in this study, the fuzzy PD turning was used to find the best performance by controlling the position and tilt-angle. Based on the fuzzy PD rules, the inputs are position and tilt angle, while the speed of the robot is output. Therefore, to study the performance of the robot, the output was tuned by the PD controller. The trial and error of K_p and K_d were implemented in this experiment. The self-balancing robot is able to control the robot effectively in real-time with some parameter changes. Finally, the result shows that the fuzzy-PD algorithm can be implemented to control the two-wheeled self-balancing robot as well as to prevent the robot from failing.

DOI: https://doi.org/10.11113/jm.v46.500

Figure 8 and Figure 9 show the relationship between the speed and the distance in a certain time. From the graph, the speed is directly proportional to the position. Based on Figure 8, the speed of the two-wheeled self-balancing robot starts to increase until 1 s and then decreases to reach stability with 8 s. Same goes for the displacement and the torque of the robot, which are returned to the point of origin in a certain time. Figure 10 shows the output of motor torque. From the graph, it has the highest torque in starting the movement, then starts to decrease until 5 s. With good dynamic performance and robustness, the device is stable throughout, demonstrating that the combustible PD control method is efficient and effective.

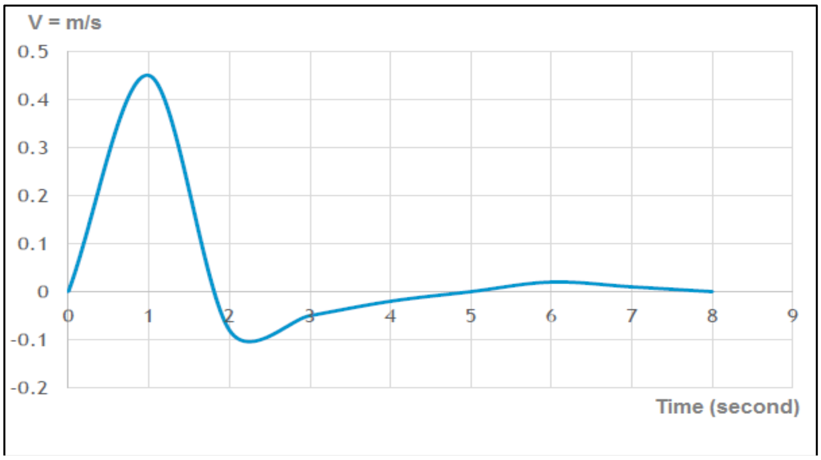


Figure 8: Speed in the second

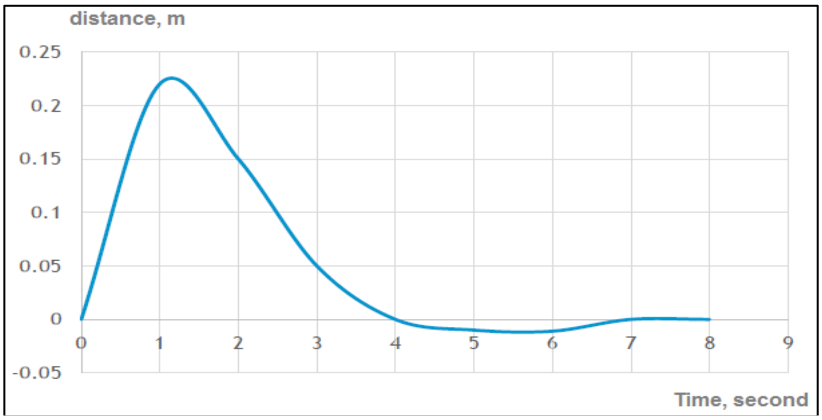


Figure 9: Distance in the second

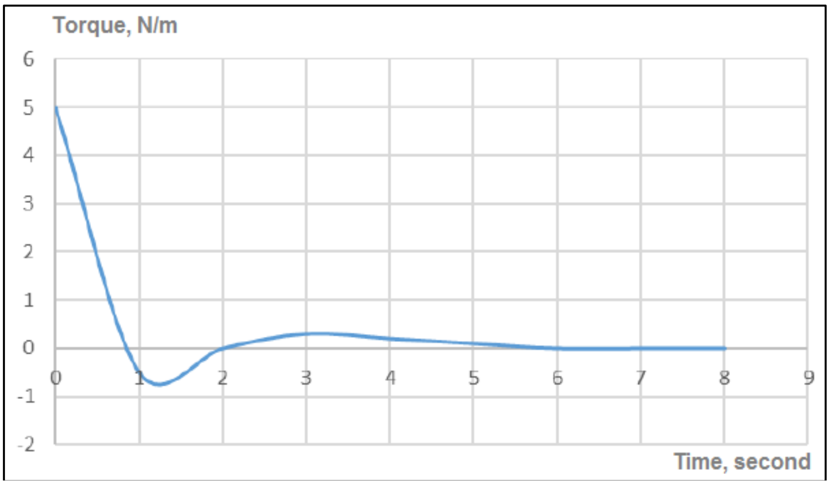


Figure 10: Torque in the second

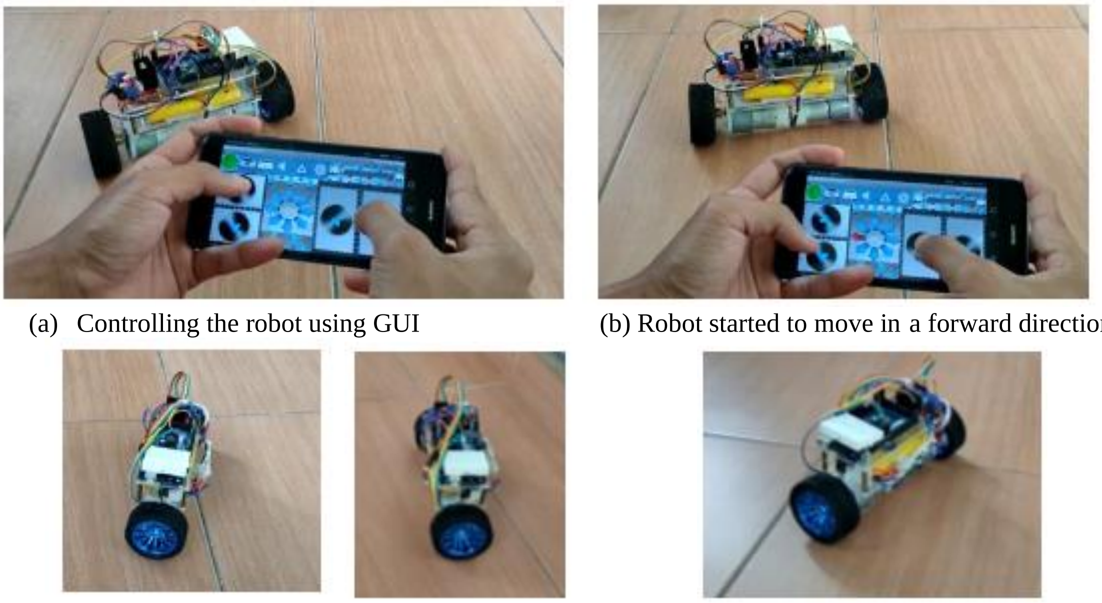
DOI: https://doi.org/10.11113/jm.v46.500

For a real implementation, the prototype of a two-wheeled self-balancing robot was studied for stability as well as balancing during the movement of the robot. This stability was seen based on the robot's ability to move and balance based on instructions that were controlled by the GUI. The algorithm from the simulation was passed to Arduino Uno using automatic code generation. The measurements of output performance were obtained by changing the controller of tilt angle and position from the IMU sensor. Table 4 shows the gain parameters of K_p and K_d for controlling and balancing the robot. In this study, the controllers were chosen based on try and error through the algorithm and implementation into Arduino programming. The choice of steady-state error is less than 5% and the maximum overshoot is about 20%.

Table 4: Gain parameter for K_p and K_d

Gain parameter	Position	Tilt angle
K_p	31.9	23.1625
K_d	0.2765	0.277

Meanwhile, Figure 10 shows the movement of the robot in four directions: - forward, reverse, turn left, and turn right by controlling the GUI. It shows that the robot was successful and able to move in an upright position and stable in all directions. In terms of speed, when the robot was turned left and right, the speed was slowed down compared to the straightforward and reverse direction. However, sometimes, the presence of the noise will make the system unstable.



(a) Controlling the robot using GUI

(b) Robot started to move in a forward direction

(c) and (d) Robot tries to stabilize its position

(e) Robot in turn left direction

Figure 10: The action of the mobile robot in 4-directions

From the observation, the robot was able to balance the robot's position during the movement at a certain time. In this work, there are five experiments done in order to study the performance of the robot. Table 5 summarizes the performance of the robot based on the capable robot's balance and speed.

Table 5: Summary of the performance balancing and speed of the two wheeled-balancing robot

Movement	Balancing	Speed
Forward direction	Able to balance	Greater speed
Backward direction	Able to balance	Greater speed

DOI: https://doi.org/10.11113/jm.v46.500

Turn left	Slightly balancing	Less speed
Turn right	Slightly balancing	Less speed

4.0 CONCLUSION

According to the outcomes, the two-wheeled self-balancing robot effectively maintains its balance and executes smooth movements based on given instructions. In this work, the robot's actions were directed through a user interface (GUI) that was developed by using Android applications. Following a phase of fine-tuning the parameters, the control of the two-wheeled self-balancing mobile robot showed a high efficiency of around 70-80% in terms of the overall control effort. The findings also demonstrate the successful application of the fuzzy PD control algorithm in achieving stable self-balancing for the two-wheeled robot. Nevertheless, some limitations emerge from the result such as the result still a slight lack of robustness, which could potentially be enhanced by implementing the fuzzy-PID concept. Additionally, it is suggested that alternative intelligent controllers could be explored and integrated with IoT for future research endeavors.

ACKNOWLEDGMENTS

The author would like to acknowledge Universiti Teknologi MARA Kampus Permatang Pauh for supporting in completing this work.

REFERENCES

1. Wael, Y., Mohammed, A., *Design and Implementation of an Experimental Seqway Model.* AIP Conference Proceedings, 2009, **1107**, p, 350 -354.
2. Shekhawat, A. S., Rohilla, Y., *Design and Control of Two-wheeled Self-Balancing Robot using Arduino*. International Conference on Smart Electronics and Communication (ICOSEC*)*, Trichy, India, 2020, pp. 1025-1030
3. Juang, H. S., Lum, K. Y., *Design and control of a two-wheel self-balancing robot using the arduino microcontroller board*, 10th IEEE International Conference on Control and Automation (ICCA), Hangzhou, China, 2013, pp. 634-639.
4. D.P.V.J. Jayakody, & K.P.G.C. Sucharitharathna. Control Unit for a Self-Balancing Prototype. *Global Journals of Research in Engineering*, *19*(J1), 2019, 7–12.
5. Celik, Y., Mahit, G., *Designing an Object Tracker Self-Balancing Robot*, Academic Platform Journal of Engineering and Science, 2018, 6-1, p. 124-133.
6. Velagić, J., Kovač, I., Panjević, A., and Osmanović, A., *Design and Control of Two-Wheeled and Self-Balancing Mobile Robot,* 2021 International Symposium ELMAR, *Zadar, Croatia*, 2021, pp. 77-82, doi: 10.1109/ELMAR52657.2021.9550938.
7. Park J.H., Cho B. K., *Development of a self-balancing robot with a control moment gyroscope*. International Journal of Advanced Robotic Systems. 2018, **15(2)**.
8. Franček, P., Jambrošić, K., Horvat, M., Planinec, V., *The Performance of Inertial Measurement Unit Sensors on Various Hardware Platforms for Binaural Head-Tracking Applications.* Sensors 2023, *23*, 872. https://doi.org/10.3390/s23020872
9. Meghji, M., *et al*., *An Algorithm for the Automatic Detection and Quantification of Athletes' Change of Direction Incidents Using IMU Sensor Data*, in *IEEE Sensors Journal*, vol. 19, no. 12, pp. 4518-4527, 15 June 15, 2019, doi: 10.1109/JSEN.2019.2898449.
10. An W., and Li, Y., *Simulation and control of a two-wheeled self-balancing robot,* 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China, 2013, pp. 456-461
11. Vicky, M., Barokatun, H., Yun, T. K. P, Okcy, S., D*esign and Simulation of Two-Wheeled Balancing Mobile Robot with PID Controller*, International Journal of Sustainable Transportation Technology Vol. 3, No. 1, April 2020, 12-19
12. Zahra, K. J., Hamed, K. S., *A New Approach for Control of Two-wheeled Mobile Robot,* Universal Journal of Electrical and Electronic Engineering 6(2): 71-78, 2019
13. Quang, V. N., Chai, Y., Trong, T. N., *The fuzzy-PID based-pitch angle controller for small-scale wind turbine***,** International Journal of Power Electronics and Drive System (IJPEDS), 2020, **Vol. 11, No. 1**, pp. 135~142.

 DOI: https://doi.org/10.11113/jm.v46.500

14. Jamil, O., Jamil, M., Ayaz, Y., Ahmad, K., *Modeling, control of a two-wheeled self-balancing robot*, 2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE), Islamabad, Pakistan, 2014, pp. 191-199
15. Mai, T. A., Dang, T. S., Anisimov, D. N. and. Fedorova, E., *Fuzzy-PID Controller for Two Wheels Balancing Robot Based on STM32 Microcontroller*, 2019 International Conference on Engineering Technologies and Computer Science (EnT), Moscow, Russia, 2019, pp. 20-24
16. T. Liu, X. Wang, H. Zhou, X. Che, H. Liu, and Q. Wang, "Design and Control of a Two-wheeled Self-balancing Robot made in 3D Printing," *2018 Chinese Automation Congress (CAC)*, Xi'an, China, 2018, pp. 1211-1216,
17. Khan, H.; Khatoon, S.; Gaur, P.; Abbas, M.; Saleel, C.A.; Khan, S.A. Speed Control of Wheeled Mobile Robot by Nature-Inspired Social Spider Algorithm-Based PID Controller. *Processes* **2023**, *11*, 1202.
18. Gong, Y., X. Wu, X., Ma, H., *Research on Control Strategy of Two-Wheeled Self-Balancing Robot*, 2015 International Conference on Computer Science and Mechanical Automation (CSMA), Hangzhou, China, 2015, pp. 281-284.
19. Prakash, K., Thomas, K., *Study of controllers for a two wheeled self-balancing robot, 2016 International Conference on Next Generation Intelligent Systems (ICNGIS)*, Kottayam, India, 2016, pp. 1-7
20. Ogata, K., Modern Control Engineering. 5th Edition, Pearson, Upper Saddle River.
21. Kim, S., Kwon, S. Dynamic modeling of a two-wheeled inverted pendulum balancing mobile robot. *Int. J. Control Autom. Syst.* **13**, 926–933 (2015).
22. Hamza, M., Zaka-ur-Rehman, Zahid, Q., Tahir, F., and Khalid, Z., *Real-Time Control of an Inverted Pendulum: A Comparative Study*, 2011 Frontiers of Information Technology, Islamabad, Pakistan, 2011, pp. 183-188, doi: 10.1109/FIT.2011.41.
23. Jin S, Ou Y., *A Wheeled Inverted Pendulum Learning Stable and Accurate Control from Demonstrations*. Applied Sciences. 2019; 9(24):5279
24. Junfeng, W., Wanying, Z., Shengda, W., *A Two-Wheeled Self-Balancing Robot with the Fuzzy PD Control Method,* Mathematical Problems in Engineering, 2012, **vol. 2012**. p. 1-13.
25. Bello, A., Olfe, K.S., Rodríguez, J., Ezquerro, J.M., and Lapuerta, V. *Experimental verification and comparison of fuzzy and PID controllers for attitude control of nanosatellites*, Advances in Space Research, Volume 71, Issue 9, 2023, Pages 3613-3630. https://doi.org/10.1016/j.asr.2022.05.055.
26. Barakat, M. *Optimal design of fuzzy-PID controller for automatic generation control of multi-source interconnected power system*. Neural Comput & Applic **34**, 18859–18880 (2022). https://doi.org/10.1007/s00521-022-07470-4
27. Mohamed, N. M. A., Abdalaziz, A. A. A., Ahmed, A. A., and Ahmed, A. A. A., *Implementation of a PID control system on microcontroller (DC motor case study)*,2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), Khartoum, Sudan, 2017, pp. 1-5.