Journal of Mechatronics and Artificial Intelligence
Homepage: http://ejournal.upi.edu/index.php/jmai/
p-ISSN 3062-729X e-ISSN 3048-4227

Designing Wheeled Robot with Rocker Bogie System for Precision Farming

Muhammad Hafizh Aditya[1], M Bima Nugraha[1,*]

[1] Universitas Multimedia Nusantara, Banten, Indonesia
*Correspondence author email: mb.nugraha@umn.ac.id

ARTICLE INFO

Article History:
Submitted/Received 14 Oct 2025
First Revised 14 Nov 2025
Accepted 15 Nov 2025
First available online 26 Nov 2025
Publication Date 01 Dec 2025

Keywords:
Autonomous Robot,
PID Control,
Precision Farming,
Rocker-Bogie System

ABSTRACT

This study presents the development of a small autonomous robot designed to support precision farming through improved mobility and control efficiency. The robot employs a Rocker-Bogie suspension system to maintain stability and traction across varied terrain, ensuring even load distribution. Six DC motors are controlled using a discrete PID system optimized through MATLAB simulations to achieve precise speed and position control. By integrating mechanical design and control optimization, the proposed system enhances operational efficiency, reduces environmental impact, and contributes to sustainable and intelligent agricultural practices.

1. Introduction

In the 20th century, the green revolution brought increased productivity through genetically improved varieties, synthetic fertilizers, chemical fertilizers, and pesticides that reduced yield loss [1], precision farming encourages farm management strategies using information technology that serves to increase production potential and reduce environmental impact [2]. The main objective of precision farming is to minimize environmental impact by reducing the use of chemicals [3]. The precision farming system can be applied to manage inputs and outputs based on quality and quantity coefficients to control production [4]. Hence the need for small and intelligent robots that can reduce waste and environmental impact and improve food security [5].

The Rocker-Bogie suspension configuration is the most popular suspension system for rover robots [6]. The Rocker-Bogie suspension design is a mobility program that has been recognized for its vehicle stability and its ability to overcome obstacles [7]. This suspension is a six-wheeled vehicle mechanism that passively keeps all six wheels in contact with the surface [8]. This mechanism uses two arms that are each mounted on a wheel. The Rocker-Bogie mechanism does not use springs. The wheels in front of this mechanism function to pass through obstacles on the path that the robot will pass [9]. This suspension mechanism uses four arms of different sizes that function as a suspension system [10]. The benefit of this mechanism is that the suspension system can also distribute the load evenly to all wheels, so that not only some wheels experience excessive loads [11]. However, according to research results, the resulting prototype is more stable when moving on hard and sloping terrain, rather than on soft terrain [12] [15]. To maintain stability, it is important to provide average body dynamics throughout the rocker-bogie suspension system. This makes it possible to maintain a rough average of the current terrain [13][14].

To run the robot, 6 DC motors are used whose direction and speed are controlled by a PID control system [16]. The main problem with DC motors is to control the angular velocity at the reference. The angular velocity can be adjusted by adjusting the voltage entering the DC motor [17]. To create a PID model, it is necessary to perform simulations. However, the results of the simulation may not be the same as the hardware implementation [18], this is due to differences in the quality and condition of the hardware that does not match the existing datasheet and is influenced by environmental conditions that are not as ideal as the conditions when simulated. The PID control that will be used in this system is not a conventional PID control that requires an additional module to interact with a digital computer, but uses the discrete PID of the DC motor speed control to easily connect the digital computer with the motor [19]. PID control is a control that will regulate Proportional Integral Derivative, by regulating these three components, it can reduce errors and provide better output when compared without using PID control [20]. Although the PID controller is an excellent controller choice for regulating the speed of a DC motor, it becomes ineffective due to load instability and adjustments in the set speed [21]. To obtain PID data, it is necessary to analyze the performance of the control system from the speed and position information on the DC motor, because each motor has different constants [22]. The PID controller used is as an algorithm to control the speed and position of the DC motor [23]. The result of PID will provide a stable dc motor and constant rotation speed by reducing rise time, settling time, overshoot and steady state error [24]. Overshoot is the maximum peak value of the response curve measured from the desired response of the system. To get good PID results, the MATLAB application is used to calculate the PID controller results so that the PID control parameter values can be obtained [25].

Therefore, this study focuses on the development and implementation of a small autonomous robot equipped with a Rocker-Bogie suspension system and a discrete PID control for DC motors to enhance stability and adaptability in precision farming applications. By integrating mechanical design and control optimization, this research aims to support sustainable agriculture through efficient, intelligent, and environmentally friendly farming operations.

2. Methods

2.1. Design Concept

The main concept of this system is to make a robot that can drive on the ground more stable and more flexible. There are 6 DC motors that will drive the motor and use the Rocker Bogie system as a suspension on this robot.

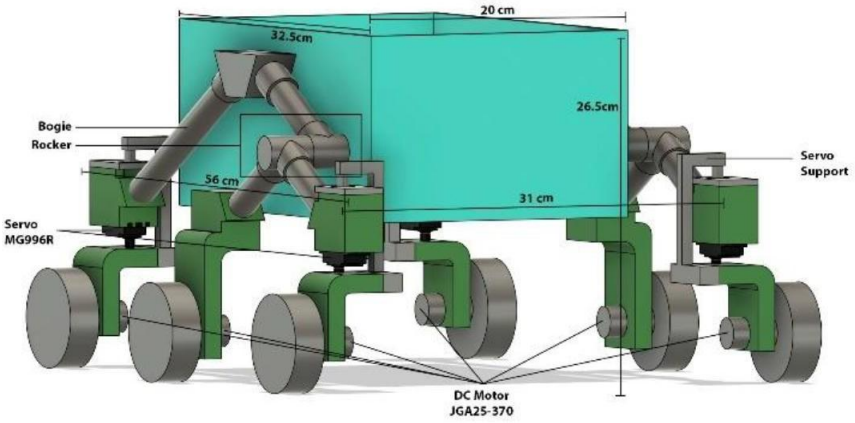

Figure 1: The proposed design of 6-wheel rover for precision farming mobile robot.

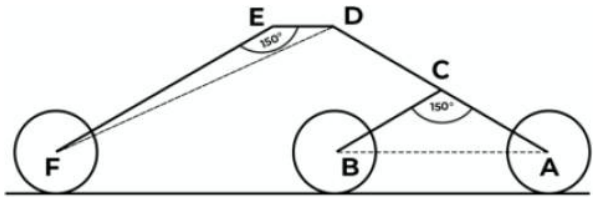

Figure 2: Schematic of Rocker-Bogie design on straight surface.

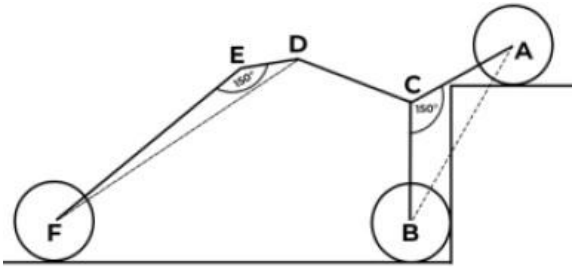

Figure 3: Schematic of Rocker-Bogie design on a stairway obstacle course.

Based on the figure 1-3, the robot has dimensions of 32.5 x 20 cm with a height of 26.5 cm. and it is known that this suspension has two different parts, namely the rocker which is the longest link at the back and the bogie which is a short link connected to the pivot that will be placed at the front. The length of the rocker is 247.585 mm, the length of the bogie is 121.739 mm, the length of the link connecting the mount to the pivot is 125.846 mm and the length of the mount is 56.785 mm.

To find out what is the maximum height that can be passed by the robot using rocker-bogie suspension and knowing the maximum distance between the rocker and bogie is to use trigonometric calculations which will be done in the equation (1):

$$AB^2 = AC^2 + BC^2 - 2 \times BC \times AC \times \cos(\alpha) \quad (1)$$

$$AB^2 = 121.739^2 + 121.739^2 - 2 \times 121.739 \times 121.739 \times \cos(120)$$

$$AB^2 = 14820.784 + 14820.784 - \left(2 \times 14820.784 \times -\frac{1}{2}\right)$$

$$AB^2 = 29640.768 - (-14820.784)$$

$$AB = \sqrt{44461.152}$$

$$AB = 210.858$$

Since the angle of the two joins is not 90°, it is necessary to calculate using the trigonometric formula and obtain the total distance between the front wheel and the center wheel is 21 cm with AC is the join of the front wheel, BC is the join for the center wheel, AB is the distance for both wheels which will be a reference for the height that can be passed by the robot. After getting the equation to find out the maximum height of obstacles that can be passed, calculations will be made to find the maximum range of the robot, it will be calculated by Equation 2.

$$DF^2 = DE^2 + EF^2 - 2 \times DE \times EF \times \cos(\beta) \quad (2)$$

$$DF^2 = 56.785^2 + 247.585^2 - 2 \times 56.785 \times 247.585 \times \cos(150)$$

$$DF^2 = 3224.536 + 61298.332 - \left(2 \times 14059.114 \times \left(-\frac{1}{2}\sqrt{3}\right)\right)$$

$$DF^2 = 64522.868 - (-24351.099)$$

$$DF = \sqrt{88873.967}$$

$$DF = 298.117$$

DF in the data above is the distance from the rear mount, DE is the distance between the two main mount points, EF is the length of the rocker join and β is the angle from point E. Based on the calculations done previously, if the obstacle is in the form of a ladder, the maximum length that can be reached by the rocker-bogie suspension is 29 cm. From the entire calculation, it can be concluded that the maximum

height that can be passed by the robot is 21 cm with the maximum length if the obstacle is in the form of a ladder is 29 cm.

2.2. Testing and Validation Method

Testing is done to find out whether the robot can move straight and whether with the speed given in the form of PWM, the robot can move as desired. There are three data taken, namely time data, distance data by using encoder as telemetry, and speed data. To get the speed, the calculation will be done with the formula:

$$Speed = \frac{Linear\ Distance}{Time}$$

After obtaining the speed data from each experimental test, the results for all repetitions are averaged to determine the mean travel time and mean speed of the robot. These averaged values are then used to analyze performance differences under various operating conditions. Specifically, three speed modes—normal, minimum, and maximum—are compared across three distinct surface types: smooth floor, corn block, and natural ground. This comparison aims to evaluate the robot's adaptability and stability on different terrains, providing insight into the effectiveness of the Rocker- Bogie suspension system and the responsiveness of the PID control in maintaining consistent motion performance.

3. Results and Discussion

There are two types of tests to analyze, namely tests for the PID system, and tests to calculate the time, speed and distance to be traveled by the robot.

3.1. PID Motors

This test is conducted to determine and select the ideal P, PI, PD and PID parameters for each DC motor. The encoder on the motor will be used as the main feedback to be a reference for determining the PID parameters on each motor. By knowing the parameters of the PID control system, the two motors can operate synchronously by generating and maintaining and harmonizing the speed in each motor.

This test is carried out using two types of applications, namely Thonny IDE and MATLAB applications. The Thonny IDE application will function to find the step response of a DC motor that has an encoder. This data will be used as a reference to find the PID parameters that will be searched using MATLAB. In the MATLAB application, the step response data that has previously been found will be plotted and the transfer function is sought. After each motor has known its transfer function, the next step is to find the PID parameters using the toolbox available in the MATLAB application. By using. After the

PID parameters are found, it is necessary to validate the manual programming in MATLAB and test on SIMULINK, this aims to recheck the results given by PID Tuning. If you have finished proving that the PID parameters given by PID Tuning are the same as the output results from manual programming and testing on SIMULINK, a simulation will be carried out to find out whether the robot can moves straight using the encoder as the main reference.

To get PID for motor motion, it is necessary to find the step response of the DC motor. To take the step response data of the motor, a motor is needed that has an encoder. Data from the encoder (pulse) will be converted into units of speed (m/s), with a reference of 1 meter = 1540 pulses, this data is obtained from direct experiments. In data collection, data will be taken for 5 seconds, and print data with an interval of 0.1 seconds. Here is the step response data from motor 3 and motor 4.

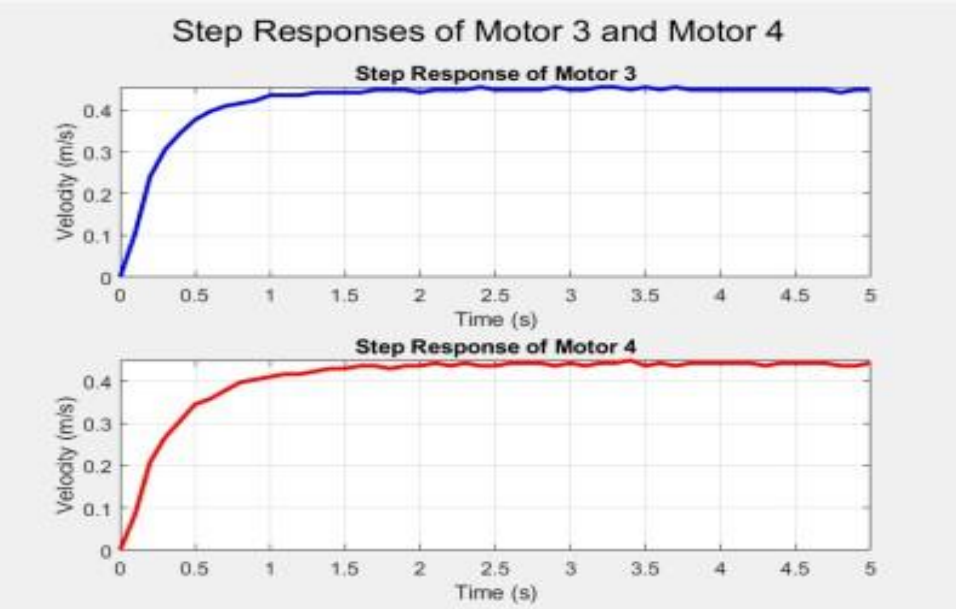

Figure 4: Comparison chart of linear speed of the two DC motors connected to the wheel.

Based on figure 4, it can be seen that the motor speed starts to stabilize at 1.3 seconds for motor 3 and 1.6 seconds for motor 4 and the motor speed starts the same at 0.44 m/s at 2.3 seconds. This occurs due to differences in the characteristics of the two motors, although the type of motor used is the same, but there are slight mechanical differences in the motor and differences in component tolerances. Therefore, these variations in data must be controlled to ensure uniform motor speed, allowing the robot to move straight forward. These two motors will be controlled by a PID control system. After the step response data is obtained, the next step is to make a comparison in the form of a plot and determine the transfer function of each motor.

After the DC motor step response data is successfully plotted, the next step is to simulate the robot based on the DC motor step response data, the following are the results of the robot simulation.

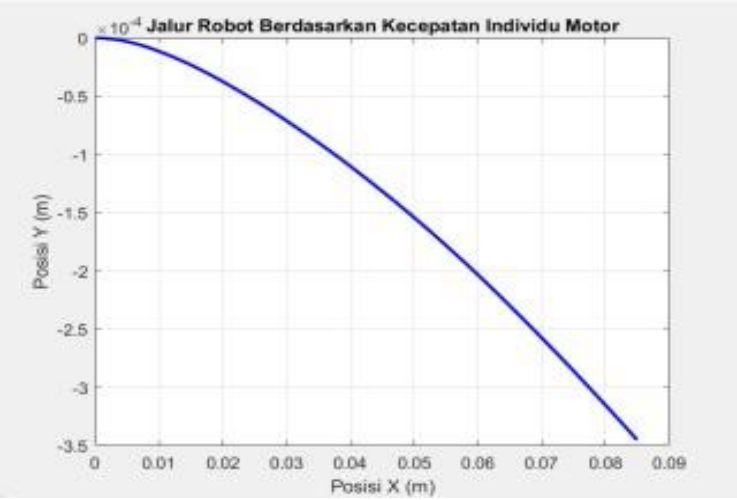

Figure 5: Simulation results of robot moving based on DC motor step response data.

The Figure 5 shows that the speed of motor 3 is faster than the speed of motor 4, this causes the robot to moves at an angle due to the difference in speed generated. This proves that, if both motors are not implemented PID, the robot will move straight.

3.2. Motor Transfer Functions

After the step response data is obtained, programming will then be carried out in MATLAB to find out each transfer function of the two motors, here are the transfer functions of the two motors.

3.2.1. Motor 3 Transfer Function

Here is the transfer function of motor 3:

$$TF = \frac{0.448}{10.2810\text{s} + 1} \tag{3}$$

After the transfer function is found, here are the plotting results of the transfer function of motor 3.

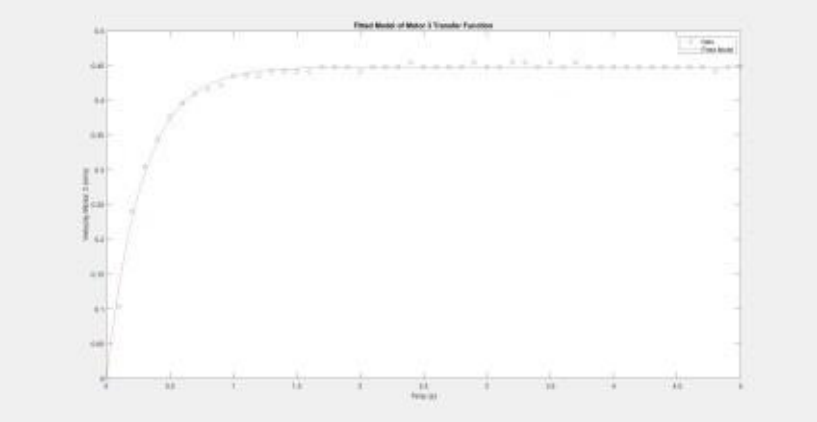

Figure 6: Motor 3 transfer function plot.

3.2.2. Motor 4 Transfer Function

Here is the transfer function of motor 4:

$$TF = \frac{0.4}{0.2810\text{s} + 1} \tag{4}$$

After the transfer function is found, here are the plotting results of the transfer function of motor 4.

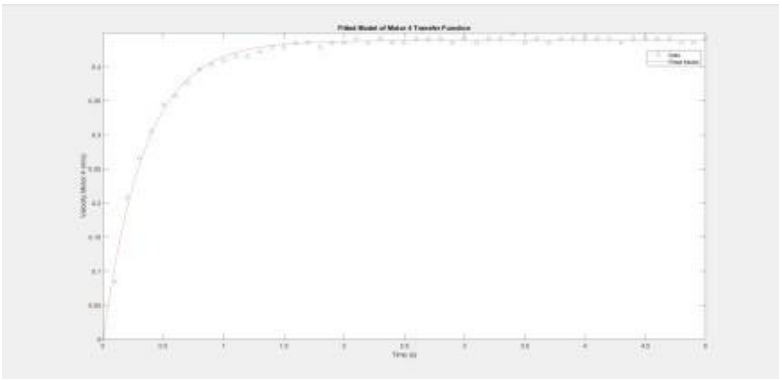

Figure 7: Motor 4 Transfer function plot.

3.2.3. Transfer Function Comparison

Based on Figure 8, the results of the transfer function plot are obtained by averaging the speed of the two motors, this data is only for comparison only and both motor 3 and motor 4 data can be averaged because the speed generated from the two motors is not too much different.

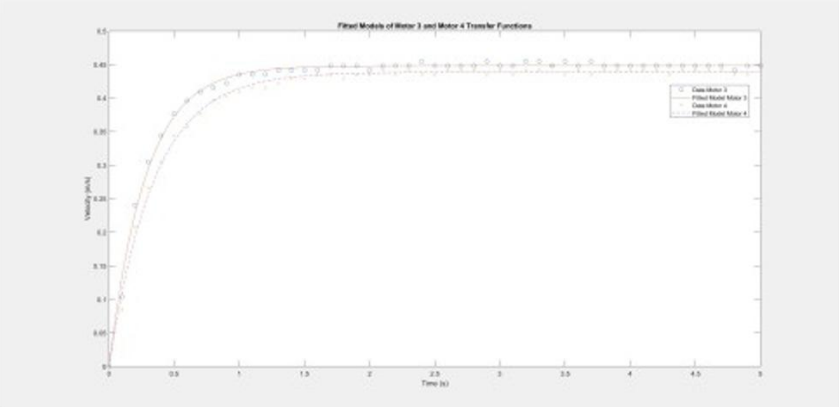

Figure 8: Comparison results of transfer function plots of motor 3 and motor 4.

3.3. PID Parameter

After each transfer function has been found, the next step is to find the PID parameters for each motor. Here are the parameters for each transfer function:

3.3.1. PID Parameters for Motor 3

In the experiment to find PID by doing PID tuning (figure 9 and figure 10), Kp and Ki were obtained, namely 13.7816 and 49.0507. In this experiment, only Kp and Ki are used because when using Kd, the data generated occurs overshoot, therefore the Kd parameter is not used. With the values of Kp and Ki obtained the rise time and settling time obtained is less than 0.2 seconds and has a peak of up to 1 which has provided ideal results for the motor.

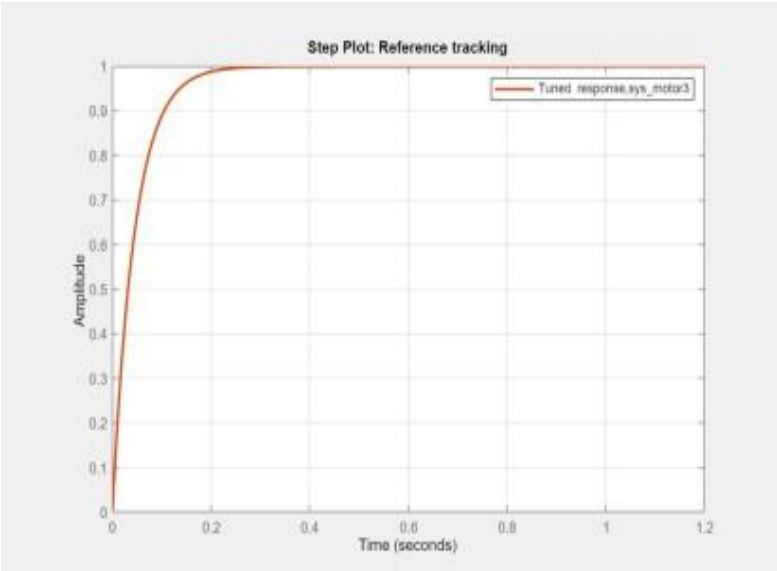

Figure 9: Motor 3 PID response graph.

Controller Parameters

	Tuned
Kp	13.7816
Ki	49.0507
Kd	0
Tf	n/a

Performance and Robustness

	Tuned
Rise time	0.1 seconds
Settling time	0.178 seconds
Overshoot	0 %
Peak	1

Figure 10: PID parameters and performance results for motor 3.

3.3.2. PID Parameters for Motor 4

Figure 11: Motor 4 PID response graph.

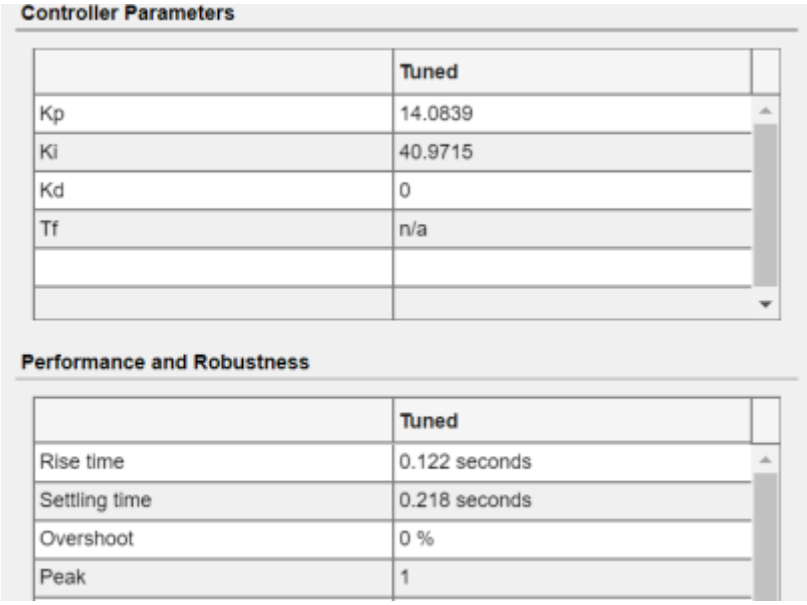

Figure 12: PID parameters and performance results for motor 4.

Based on the data contained in Figure 11 and Figure 12, the parameters used are still the same, namely Kp and Ki only, because when using Kd, it still gives overshoot. The output obtained from the parameters that have been found is that the overshoot has reached 0% and managed to reach peak 1 without steady state error. The results given are good because the rise time and settling time become faster than without using a PID control system.

3.3.3. Parameter Testing

After finding the PID parameters of the two motors, testing was carried out to implement the PID parameters into the system. This test will be carried out on three different types of surfaces, namely, floor surface, concrete surface and ground surface. This test does not rely solely on the encoder to correct the forward direction, but uses the MPU6050 IMU sensor. The data used from the MPU6050 sensor is pitch data and yaw data. The yaw data will be used as reference for the robot to moves straight and the pitch data will be used as reference for the robot to maintain speed.

a. Experiments on Floor Surface

From the results of the tests carried out, experiments were carried out to see the results of the step response data from the two motors and then tested to obtain travel data in the form of encoder data, time, speed and final reading of yaw data. In the first test, namely testing on the floor surface, from Figure 13 it can be seen that the speed has stabilized at 0.44 m/s. This proves that the implementation of PID on the robot has been successful. However, when comparing the PID simulation results with the PID implementation results, it can be seen that the rise time in the PID implementation on the robot is slower

than the rise time in the simulation. This is due to two main factors, namely the mass of the robot and the speed of the other four motors which are not the same as motors that have encoders. The rise time in this implementation is 0.68s.

After obtaining step response data, an experiment was conducted to obtain travel data from the robot on the floor surface. From the experiment, it can be seen that the robot has successfully moved straight based on yaw data readings and encoder readings that have successfully moved 5 meters. The speed generated by the robot in moving 5 meters has successfully reached the setpoint of 0.44 m/s.

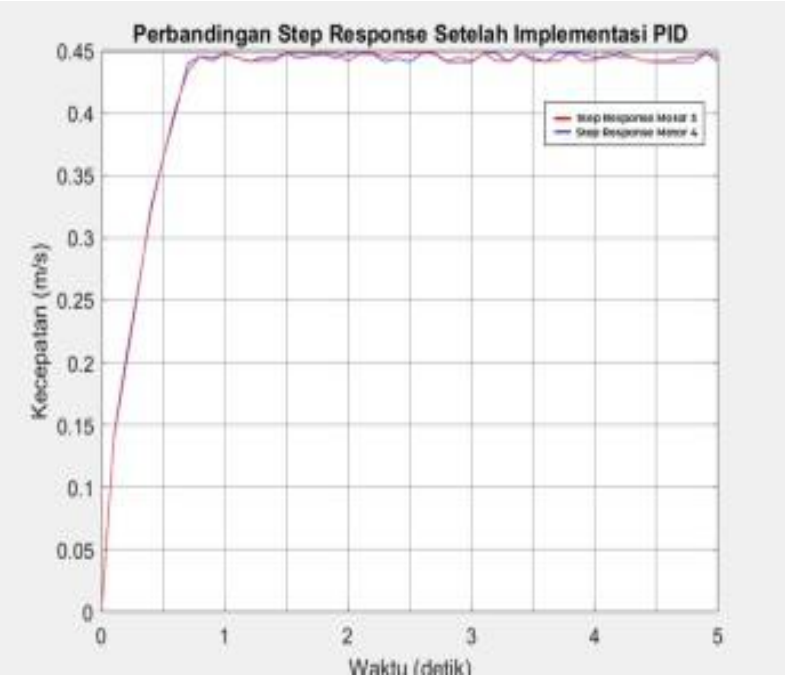

Figure 13: Step response graph of PID implementation on floor surface.

Table 1: PID Implementation Testing Data on Floor Surface.

Encoder Right (Pulse)	Encoder Left (Pulse)	Time (s)	Speed (m/s)	Yaw
7766	7791	11,21	0,44603	0,45
7784	7750	11,37	0,439754	-0,32
7707	7788	11,28	0,443262	0,76
7759	7711	11,34	0,440917	-0,89
7713	7799	11,24	0,44484	0,12
7799	7730	11,35	0,440529	0,56
7753	7767	11,33	0,441306	-0,15
7715	7735	11,26	0,44405	0,97
7704	7741	11,3	0,442478	-0,45
7794	7709	11,35	0,440529	0,23

b. Experiment on Corn Block Surface

After conducting experiments on the floor, experiments will be carried out on the surface of the concrete block. From Figure 14, it can be seen that the results of the step response of motors 3 and 4 managed to stabilize at 0.44 m/s, with a rise time of 0.84s.

The next experiment is to conduct an experiment to obtain movement data from the robot move on the surface of the concrete block. From the experiments that have been conducted, it can be seen that the experiment to run the robot on the corn block surface has succeeded in making the robot moves straight, but some data shows that the final reading of the yaw data shows the number 1 and the number -1. Unlike the floor where speed correction is easier, on a concrete surface that has uneven contours, has a slight gap between concrete blocks and also a rough surface makes it a little more difficult for the motor to correct the forward direction. Also in this experiment it can be seen that the resulting speed is not stable at 0.44 m/s, but there is still a lot of data at 0.43 m/s. This is due to the types of differences that the robot goes through.

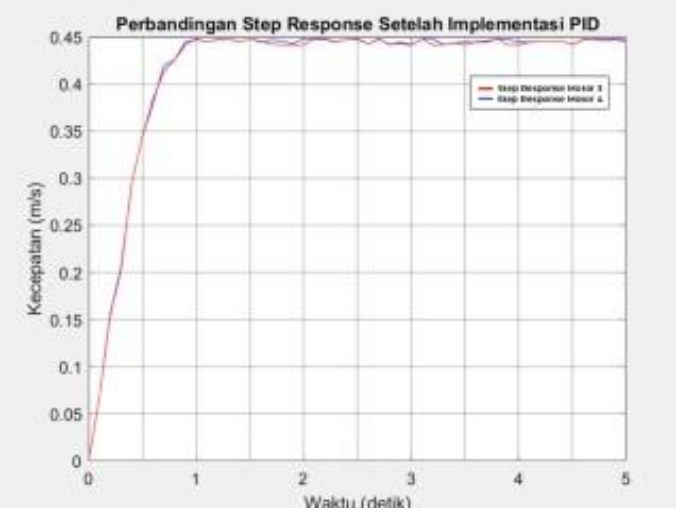

Figure 14: Step response graph of PID implementation on corn block surface.

Table 2: PID implementation testing data on corn block surface.

Encoder Right (Pulse)	Encoder Left (Pulse)	Time (s)	Speed (m/s)	Yaw
7775	7749	11,31	0,442087	0,34
7727	7736	11,35	0,440529	-1,58
7821	7810	11,42	0,437828	0,76
7785	7793	11,42	0,437828	-0,45
7843	7835	11,38	0,439367	1,89
7760	7742	11,33	0,441306	-0,12
7758	7777	11,36	0,440141	1,67
7811	7806	11,42	0,437828	-0,98
7725	7738	11,34	0,440917	0,12
7804	7815	11,41	0,438212	-0,33

c. Experiment on Dirt Surface

After conducting experiments on the corn block, experiments will be conducted on the ground surface with a slope angle of 5° to 7°. In the step response data collection experiment, it can be seen from Figure 15 that the robot managed to stabilize at a speed of 0.44 m/s, with a rise time of 0.97 s. The next step is to test the movement of the robot on the ground.

The next step is to test the movement of the robot on the ground. From the experiment that have been conducted, it can be seen that the robot managed to fix the forward direction but just like the experiment on the corn block surface, the robot could not fix the forward direction perfectly like on the floor surface. This is because the uneven surface makes it more difficult for the robot to correct its forward direction. However, the robot can still maintain a forward direction of less than 4°. In this experiment the robot still managed to maintain speed with a final speed of 0.43 m/s to 0.44 m/s.

From the experiments that have been carried out, it can be seen that the robot manages to moves straight on the floor surface well, but in experiments on the concrete surface and the ground surface, the robot turns slightly to reach more than 3 ° but not up to 4 °. In experiments on all three surfaces, it can also be seen that the robot managed to maintain speed well, although in the ground experiment, some experimental results only reached 0.43 m / s. This speed difference is due to uneven surfaces. The difference in speed is because the uneven surface makes it more difficult for the robot to maintain speed perfectly. However, the resulting speed output is only 0.01 m/s different from the specified set point of 0.44 m/s. From the data above, it can also be seen that the rise time of the experiment on each surface shows different results with the floor surface with the fastest rise time and the ground with the lowest rise time. This is due to differences in the surface traveled. On the floor surface, the surface traversed is flat and not rough, while on the surface of the block and the ground both have uneven contours and have rough surfaces.

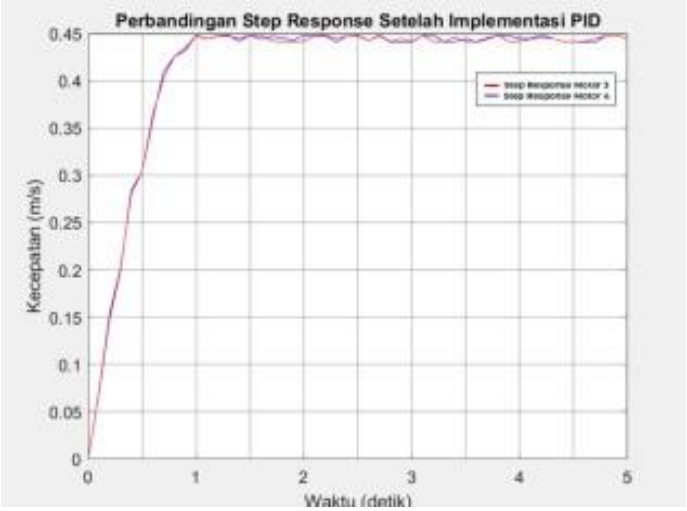

Figure 15: Step response graph of PID implementation on dirt surface.

Table 3: PID implementation testing data on dirt surface.

Encoder Right (Pulse)	Encoder Left (Pulse)	Time (s)	Speed (m/s)	Yaw
7724	7733	11,35	0,440529	2,08
7701	7781	11,47	0,43592	0,53
7765	7712	11,46	0,4363	-1,26
7748	775	11,36	0,440141	2,99
7789	772	11,34	0,440917	1,98
7718	7762	11,42	0,437828	0,32
7754	7731	11,36	0,440141	1,41
7729	7745	11,45	0,436681	3,14
7740	7755	11,35	0,440529	1,44
7736	7774	11,37	0,439754	0,23

4. Conclusion

Based on the tests that have been carried out, it can be concluded that the step response data from the two motors has successfully become a reference for finding the right transfer function to find PID parameters. After the PID parameters are found, two types of tests are carried out to prove the results of the PID parameters are suitable for the two motors, it can be seen that the PID has matched the existing transfer function through testing using code in MATLAB and by using SIMULINK. The PID obtained has also succeeded in making the robot maintain speed with a predetermined set point and can make the robot moves straight with the help of PID and the MPU6050 IMU sensor. On the surface of the block and the ground, the robot can move straight and maintain speed, but not as well as on the floor surface with a slight error below 4° and a speed error of 0.01 m/s.

References

[1] R. Finger, S. M. Swinton, N. E. Benni, A. Walter, “Precision Farming at the Nexus of Agricultural Production and the Environment”, Annual Review of Resource Economics, vol. 11, 2019. https://doi.org/10.1146/annurev-resource-100518-093929.

[2] Y. Vecchio, M. D. Rosa, F. Adinolfi, L. Bartoli, M. Masi, “Adoption of precision farming tools: A context-related analysis”, Land Use Policy, vol. 94, 2020. https://doi.org/10.1016/j.landusepol.2020.104481.

[3] M. Fawakherji, C. Potena, A. Pretto, D. D. Bloisi, D. Nardi, “Multi-Spectral Image Synthesis for

Crop/Weed Segmentation in Precision Farming', Robotics and Autonomous Systems, vol. 146, 2021. https://doi.org/10.1016/j.robot.2021.103861.

[4] E. Symeonaki, K. Arvanitis, D. Piromalis, "A Context-Aware Middleware Cloud Approach for Integrating Precision Farming Facilities into the IoT toward Agriculture 4.0", Applied Agri-Technologies, 2020. https://doi.org/10.3390/app10030813.

[5] R. F. Carpio, C. Potena, J. Maiolini, G. Ulivi, N. B. Rosello, E. Garone, A. Gasparri, "A Navigation Architecture for Ackermann Vehicle in Precision Farming", IEEE Robotics and Automation Letters, vol. 5, no. 2, 2020. https://doi.org/10.1109/LRA.2020.2967306.

[6] S. Noble, K. K. Issac, "An improved formulation for optimizing rocker-bogie suspension rover for climbing steps," Mechanical Engineering Science, vol. 223, no. 18, 2019. https://doi.org/10.1177/0954406219863086.

[7] Shivam, J. A. Kumar, P. Subramaniam, L Mercy, D. Raj, "Design and fabrication of smart rover using rocker bogie mechanism", AIP Conference Proceedings 2311, 050003 (2020). https://doi.org/10.1063/5.0034190.

[8] P. B. Saraiya, "DESIGN OF ROCKER BOGIE MECHANISM" International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 08, Aug 2020. https://www.irjet.net/archives/V7/i8/IRJET-V7I8255.pdf.

[9] P. Patil, S. Bhokardole, D. Bhandarkar, "The Rocker Bogie Mechanism: Design and Fabrication", International Journal of Innovations in Engineering and Science, Vol. 6 , No. 10, 2021, PP. 76 – 79. https://doi.org/10.46335/IJIES.2021.6.10.16.

[10] S. Seralathan, A. Bagga, U. K. Ganesan, V. Hariram, T. Micha Premkumar, S. Padmanabhan, "Static structural analysis of wheel chair using a rocker bogie mechanism", Materials Today: Proceedings, Volume 33, Part 7, 2020. https://doi.org/10.1016/j.matpr.2020.05.658.

[11] Z. Zainol, M. . Rafeeq, S. F. Toha, and A. S. Idris, "Locomotion Performance of Amphibious Robot Vehicle using Transformable Rocker-bogie Mechanism", Elektrika, vol. 20, no. 2-3, pp. 111–116, Oct. 2021.

[12] H. B. Khan, A. Wali, S. Masroor, "Design and Modification of Arduino-Based Rocker Bogie Mechanism for Security Purposes", Balochistan Journal of Engineering & Applied Sciences (BJEAS) – (p-ISSN: 2518-2706), Volume 5, No. 1.

[13] S. Panneerselvam, D. Bhoopesh, K. Akhileswaran, B. Charan, N. Aravindhan, "Design and

Fabrication of Rocker-Bogie Pesti-Bot Using Rotary Mechanism", 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS). https://doi.org/10.1109/ICACCS51430.2021.9441751.

[14] T. Pietrzyk, T. Valascho, "Implementation of a Robotic Rocker-Bogie Prototype", 2020 NDIA Ground Vehicle Systems Engineering And Technology Symposium, Autonomy, Artificial Intelligence & Robotics (AAIR) Technical Session, August 11-13, 2020.

[15] C. Cosenza, V. Niola, S. Pagano, S. Savino, "Theoretical study on a modified rocker-bogie suspension for robotic rovers", Cambridge University Press, 29 May 2023. https://doi.org/10.1017/S0263574723000656.

[16] Y. S. Kale, S. V. Rathkanthiwar, B. Sharma, J. Ghuguskar, K. Deshmukh, R. Kate, P. Thakre, "Autonomous Terrain Vehicle", INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY, 19.07.2021. https://doi.org/10.1109/iciee49813.2020.9277.

[17] A. Ma'arif, N. R. Setiawan, "Control of DC Motor Using Integral State Feedbackand Comparison with PID: Simulation and Arduino Implementation", Journal of Robotics and Control (JRC), Volume 2, Issue 5, September 2021. https://doi.org/10.18196/jrc.25122.

[18] A. Ma'arif, Iswanto, N. M. Raharja, P. Aditya Rosyady, A. R. Cahya Baswara and A. Anggari Nuryono, "Control of DC Motor Using Proportional Integral Derivative (PID): Arduino Hardware Implementation," 2020 2nd International Conference on Industrial Electrical and Electronics (ICIEE),Lombok, Indonesia, 2020, pp. 74-78, https://doi.org/10.1109/ICIEE49813.2020.9277258.

[19] S. Balamurugan and A. Umarani, "Study of Discrete PID Controller for DC Motor Speed Control Using MATLAB," 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 2020, pp. 1-6, https://doi.org/10.1109/ICCIT-144147971.2020.9213780.

[20] S. Oladipo, Y. Sun. & Z. Wang, "Optimization of pid controller with metaheuristic algorithms for dc motor drives: Review", International Review of Electrical Engineering, vol. 15, no. 5, pp. 352-381. https://doi.org/10.15866/iree.v15i5.18688.

[21] A. Zahir, Alhady, S., Wahab, A., & Ahmad, M. (2020). Objective functions modification of GA optimized PID controller for brushed DC motor. International Journal of Electrical and Computer Engineering (IJECE), 10(3), 2426-2433. https://doi.org/10.11591/ijece.v10i3.pp2426-2433.

[22] Y. B. Koca, Y. Aslan and B. Gökçe, "Speed Control Based PID Configuration of a DC Motor for

An Unmanned Agricultural Vehicle," 2021 8th International Conference on Electrical and Electronics Engineering (ICEEE), Antalya, Turkey, 2021, pp. 117-120, https://doi.org/10.1109/ICEEE52452.2021.9415908.

[23] M. Shouran and M. Habil, "Tuning of PID Controller Using Different Optimization Algorithms for Industrial DC Motor," 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2021, pp. 756-759, https://doi.org/10.1109/ICACITE51222.2021.9404616.

[24] D. Saputra, A. Ma'Arif, H. Maghfiroh, P. Chotikunnan, "Design and Application of PLC-based Speed Control for DC Motor Using PID with Identification System and MATLAB Tuner", International Journal of Robotics and Control Systems Vol. 3, No. 2, 2023, pp. 233-244 ISSN 2775-2658. https://doi.org/10.31763/ijrcs.v3i2.775.

[25] R. Rikwan, & A. Ma'arif. “DC Motor Rotary Speed Control with Arduino UNO Based PID Control”, Control Systems and Optimization Letters, Vol. 1 No.1, pp. 27-31. https://doi.org/10.59247/csol.v1i1.6.