Velocity Control System for Basketball Throwing Mechanism on the Barelang-V Robot

Rikardo Harefa[1], Rifqi Amalya Fatekha[2], Rifky Afriza[3], Senanjung Prayoga[4]

[1]Departement of Electrical Engineering,Robotics Engineering Technology Study Program,
Politeknik Negeri Batam, Batam Indonesia

[*]Email: rifqi@polibatam.ac.id

***Abstract*— The design of a control system for a basketball-shooting mechanism was conducted as part of the preparation for the 2025 ABU Asia-Pacific Robot Contest (ABU ROBOCON) with the theme "Basketball Robot." This robot is designed to perform various tasks, one of which is shooting a ball into the hoop. To carry out this task, a reliable and stable control system is required to ensure that every shot hits the hoop accurately. Shooting the ball into the hoop was tested without a control system, but the results showed poor system performance. One motor tended to overshoot the target during testing, while the other motor did not fully reach the setpoint or consistently remained below it. This is evident from the steady-state error of Motor 1 (1.22%) and Motor 2 (-0.7%). The application of PID control combined with manual tuning in this case is one of the best methods for addressing system instability, and has proven effective, with a rise time of 1 second and a stabilization time of 1.4 seconds. Similarly, the steady-state error has been significantly reduced to 0.005% for Motor 1 and 0.095% for Motor 2. Overshoot was also observed at relatively low percentages: 0.94% for Motor 1 and 1.54% for Motor 2. This study demonstrates that the use of PID control with manual tuning increases the percentage of balls entering the ring.**

Keywords: PID Control, Thrower Basketball, PWM

I. Introduction

Barelang-V is a robotics team under the auspices of the Batam State Polytechnic that focuses on robotics research and development. The team consistently trains and prepares its members to participate in national-level robotics competitions every year, including the ABU Indonesia Robot Contest (KRAI). ABU Indonesia Robocon is a branch of the Asia-Pacific Broadcasting Union (ABU) Robocon, which focuses on the development of competition-ready robots among universities in the Asia-Pacific region. Each country in Asia may send one representative team to participate in this competition. The theme for Robocon 2025 is "Basketball Robot" [1]. This competition features teams, each consisting of two robots. The robots work together as effectively as possible to score points by throwing a basketball into a hoop 2.43 meters high. The opposing team's robots are tasked with defense, where they attempt to prevent the attacking robots from scoring points in their hoop. The same applies if our robots are in the defensive position.

To perform one of the robot's tasks—throwing a ball into a hoop—a ball-launching mechanism is used that employs two ODrive DC motors as the primary actuators, which are directly connected to two vertically mounted auxiliary wheels. Pulse signals are sent to each ODrive BLDC [2], using an ESC as the driver and a Teensy 4.1 as the microcontroller

To ensure the ball is thrown precisely into the hoop, a stable and accurate ball-throwing control system is required, specifically using PID control by determining the values of Kp, Ki, and Kd [3], as well as through manual PID tuning [4] to stabilize the motor speed. In addition to PID control, testing was also conducted using PWM control to throw the ball into the hoop. The main parameters used in this ball launcher include: ring height, ball launcher height, and the distance from the ball launcher to the ring.

This study aims to apply control methods to perform basketball throws by determining a stable throwing speed in the launcher mechanism. Through this study, it is hoped that optimal and stable motor speed values for the launcher will be obtained so that it can be used effectively in robotic systems.

II. Method

The control system for this ball-throwing machine employs two main approaches: first, an open-loop system that uses only a PWM (Pulse Width Modulation) signal to drive the motor [5], and second, PID control with Kp, Ki, and Kd parameters determined using the Ziegler-Nichols method, which are then fine-tuned through manual adjustment to achieve a stable and optimal motor speed for throwing the ball into the hoop.

A. *Throwing Parameter*

The following image shows the parameters used to shoot a basketball into the hoop.

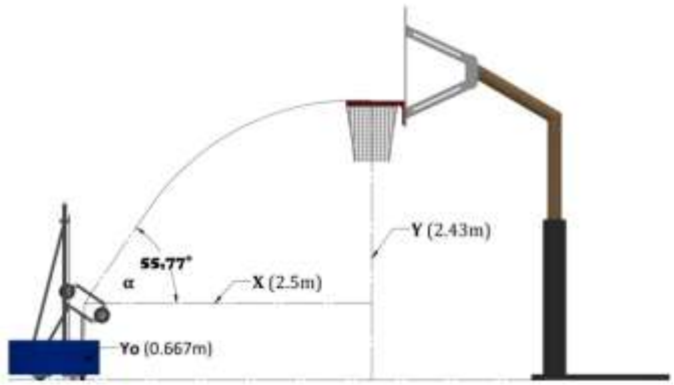

Fig 1. Baketball throwing parameters

Notes:

X = Distance from the basketball shooter to the hoop
Y = Height of the hoop
α = Angle of the basketball shot
Yo = Height of the basketball shooter

In Fig 1, the distance from the robot or basketball-shooting mechanism to the hoop is 2.5 meters (X). α is the shooting angle, which is 55.77°, a fixed angle for the shooter. Y is the height of the basketball hoop, which is 2.43 meters, and Yo is the shooter's height above the floor, which is 0.667 meters.

B. Robot Design and Mechanism

The design and mechanics of this robot comply with the Abu Robocon 2025 regulations. The throwing mechanism uses an Odrive motor and V-shaped wheels to grip the ball, ensuring that the friction from the wheels is applied precisely to the ball's surface [6]. The wheels were 3D-printed using PLA material with a 30% infill rate.

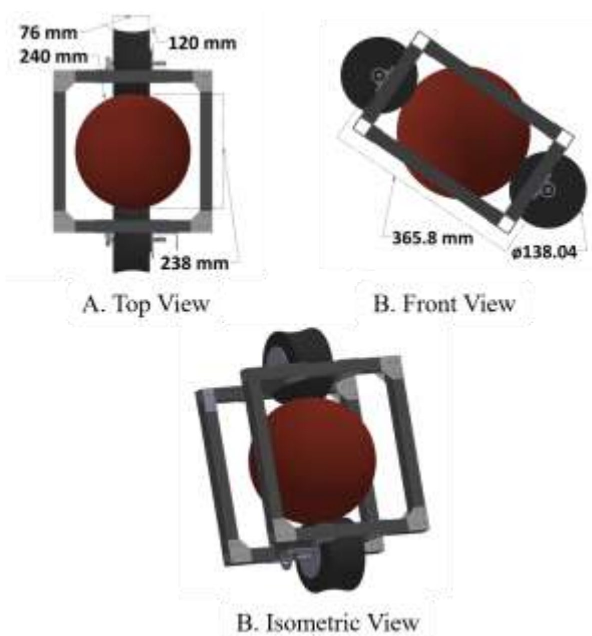

Fig 2. Top, side, and isometric views of a basketball launcher

In Fig 2, the ball-throwing mechanism is positioned at an angle of 55.77°, determined based on experiments conducted from a distance of 2.5 meters. The results of these experiments show that the shot can reach the rim, so this can serve as a reference for achieving good control when shooting a basketball.

C. DC Motor

The DC motor used in this basketball launcher is an ODrive Robotics D5065 270 kV brushless DC (BLDC) motor, controlled using an ESC driver and a Teensy 4.1 as the controller. A BLDC (Brushless Direct Current) motor is a direct current motor that does not use mechanical brushes; instead, commutation is performed electronically by adjusting the current in the stator windings based on the rotor's position, thereby generating a rotating magnetic field that drives the rotor.

Fig 3. Brushless DC Motor ODrive Robotics D5065

Fig 3 shows the ODrive motor, which serves as the main actuator in the basketball-shooting mechanism and is used to generate controlled rotational motion based on the output signal from the PID controller.

TABLE I
ODRIVE D5065 BLDC MOTOR SPECIFICATIONS

Voltage	16 – 48 V
Current	0 -70 A
Velocity	5760 RPM
Phase Resistance (mOhm)	39 mOhm
Torque	3.86 Nm
Mass	890 g
Encoder frequency (base speed)	196.61 kHz
Force	404.15 N

TABLE I lists the specifications of the Odrive D5065 motor, which is used as the main actuator in this basketball-throwing mechanism.

D. Motor Controller

The motor controller functions as a power regulator to control the motor's speed and direction of rotation according to control signals from the control system. In this basketball shooter control system, the motor controller is used to ensure the motor operates stably and responsively during the shooting process. The controller used is the Skywalker 50a Ubec 5a Hobbywing, which is an Electronic Speed Controller (ESC) for brushless DC motors. This ESC receives PWM signals from the microcontroller and regulates the amount of power delivered to the motor.

Fig 4. Hobbywing Skywalker 50a Ubec 5a Driver

Fig 4 shows the ESC controller used to regulate the motor power in this basketball-shooting device. The following are the specifications for the Hobbywing Skywalker 50A UBEC 5A ESC controller:

TABLE II
SPECIFICATIONS OF THE HOBBYWING SKYWALKER 50A UBEC 5A DRIVER

Input Voltage	2-4S Lipo, 5-12 cells NiMH
Output	Continuous 50A, Burst 65A up to 10 seconds
BEC	5A / 5V Switch mode BEC
Max Speed	210000rpm for 2 Poles BLM, 70000rpm for 6 poles BLM, 35000rpm for 12 poles BLM. (BLM = Brushless Motor)
Size	65mm*35mm*14mm
Weight	41g

TABLE II shows the specifications of the Hobbywing Skywalker 50A UBEC 5A driver, which supports BLDC motors. In addition, this driver is equipped with a safety feature regardless of the throttle lever position, the motor will not spin once the battery is connected as well as a throttle calibration function, which allows the throttle range to be calibrated for compatibility with various types of transmitters. This feature enhances safety during the initial motor calibration process.

E. PID Controller

A PID controller has three main control components: proportional (P), integral (I), and derivative (D), which are used either simultaneously or separately, depending on the desired response of a system [7]. In this study, a Proportional-Integral-Derivative (PID) controller is used to improve the accuracy and stability of the output signal when reaching the setpoint. The mathematical equation for the PID controller is expressed as follows:

$$u(t) = Kp\, e(t) + Ki \int_0^t e(t)\, dt + Kd\, de(t)/dt$$

Notes:

$u(t) =$ Output signal at time t
$e(t) =$ Error signal at time t
$Kp =$ Proportional gain
$Ki =$ Integral gain
$Kd =$ Derivative gain
$\int_0^t e(t)\, dt =$
Integral of the error signal with respect to time
$de(t)/dt =$
Derivative of the error signal with respect to time

PID controller consists of three main components: proportional, integral, and derivative [8]. The proportional component responds directly to the error; the integral component is used to eliminate steady-state error; and the derivative component improves system stability by damping sudden changes in the error. The combination of these three components is used to achieve a fast and stable system response [9]. The PID controller is implemented in a closed-loop control system, where the control signal is generated based on the difference between the reference input and the measured output. The following is a block diagram of the closed-loop system applied to this basketball shooter control system.

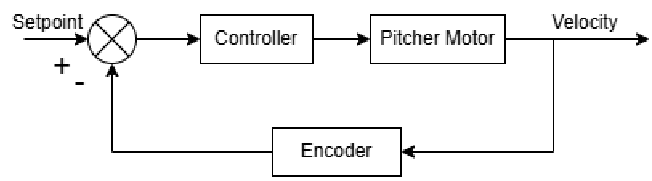

Fig 5. Closed Loop Diagram Block

Figure 5 shows a block diagram of a closed-loop control system that uses a PID controller. The setpoint value—motor speed (rev/s)—is compared with the actual speed read by the rotary encoder to generate an error signal. This error is processed by the PID controller implemented on a microcontroller to generate a PWM control signal that is fed to the brushless motor. The motor speed is then fed back to the system via the encoder sensor as closed-loop feedback. Based on the closed-loop control system shown in Figure 5, the PID controller is used to generate a control signal based on the difference between the setpoint value and the output signal. The performance of the PID controller is greatly influenced by the constants Kp, Ki, and Kd [10]. Therefore, the PID controller parameters in this control system are tuned using the Ziegler-Nichols method.

TABLE III
THE ZIEGLER NICHOLS (ZN) FORMULA

Controller	Kp	Ki	Kd
P	0.5 Kcr	∞	0
PI	0.45 Kcr	Pcr/1.2	0
PID	0.6 Kcr	Pcr/2	0.125 Pcr

In TABLE III, Kcr refers to the proportional gain (Kp) when oscillations occur in a steady state. As Kp increases from zero, the system begins to oscillate. Kcr is the value of Kp at the moment the oscillations become steady, and at that moment the system is at the boundary between stability and instability. The value of Kcr is used as a reference to determine the optimal value of Kp to achieve the desired response. Meanwhile, Pcr is the oscillation period during which the system is at the boundary between stability and instability. In some formulations, Pcr is also known as the transition time or critical oscillation time. Pcr is measured from the peak of one oscillation to the peak of the next and indicates how quickly the system responds to changes.

The performance of a control system is evaluated using several time-response parameters commonly used in control system analysis. These parameters are used to calculate rise time, stabilization time, steady-state error, overshoot, and decay conditions when throwing a ball into a hoop.

$$SS\ \text{Error} = \frac{Setpoint - Value\ Steady\ State}{Setpoint} 100\% \tag{1}$$

A steady-state error (SS Error) is the difference between the setpoint value and the steady-state signal value or the actual output value [11]. The setpoint is the reference value or the desired target value. There is also a condition known as overshoot, which is the highest output value of the system's response that exceeds the setpoint value. The following is the equation for overshoot:

$$Overshooot = \frac{Max\ Overshoot - Value\ Setpoint}{Setpoint} 100\% \tag{2}$$

In addition to steady-state error and overshoot, there is also a drop condition that occurs when shooting a ball at a hoop [12]. The drop percentage can be calculated using the following equation:

$$Drop = \frac{Steady\ State\ Value - Drop\ Minimum}{Steady\ State\ Value} 100\% \tag{3}$$

This drop, or load disturbance, is a condition where the response drops drastically when shooting a ball at the hoop [13]. The minimum drop is the lowest response value when shooting a ball at the hoop.

III. Result and Discussion

In this test, the first experiment was conducted without a control system; subsequently, the test utilized a PID control system to determine the optimal and most accurate shot at the hoop. This test required 5 basketballs, which were thrown from a distance of 2.5 meters, with the hoop at a height of 2.43 meters and the thrower standing at a height of 0.677 meters. The rotational speed of each motor was measured by an encoder mounted on it.

A. System Testing Without Control

The purpose of this initial test was to observe the motor's response without any control. This experiment was conducted by inputting a value of 35 Rps (revolutions per second). This Rps value was obtained from direct experimentation, combined with an angle of 55.77°. The test results showed that the ball reached and entered the hoop. In tests using values below 35 Rps, the basketball did not reach the hoop; similarly, when using values above 35 Rps, the response tended to be too fast, causing the ball to reach the hoop but bounce back out because the Rps value was too high. The graph below shows the uncontrolled system's response based on the encoder readings, as shown in Fig 6.

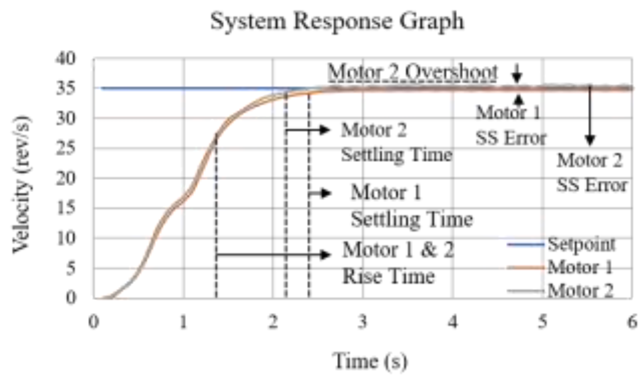

Figure 6. System response graph without PID control

In Fig 6, the system response graph shows the rise time (Tr), stabilization time (Ts), and steady-state error (ESS). The rise time is the time required for the motor response to reach 10–90% of the setpoint value. The stabilization time is the time required for the motor output to reach and remain within the setpoint range. The stabilization time threshold is typically 2% or 5%, depending on the motor response; here, a 2% threshold is used [14].

From Fig 6, it can be seen that the responses of Motors 1 and 2 reach the setpoint with a rise time of 1.3 seconds. Motor 1 reaches its stabilization time in 2.4 seconds, while Motor 2 reaches its stabilization time in 2.1 seconds. To calculate the steady-state error and overshoot using Equations (1) and (2), the following values are obtained:

$$Error\ SS\ Motor\ 1 = \frac{35 - 34.57}{35} 100\% = 1.22\% \tag{4}$$

$$Error\ SS\ Motor\ 2 = \frac{35 - 35.25}{35} 100\% = -0.7\% \tag{5}$$

$$Overshoot\ Motor\ 2 = \frac{35.53 - 35}{35} 100\% = 1.51\% \tag{6}$$

The response curve for Motor 1 remains below the setpoint, so there is no overshoot. Following the results of the motor response test without control, a ball-throwing test was then conducted. The previous response values were used for throwing the ball into a ring 2.5 meters away. The following are the results of the ball-throwing test into the ring without control:

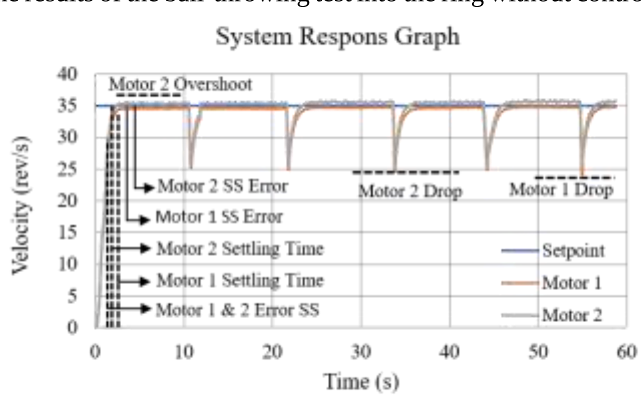

Fig 7. System response graph without control when throwing a ball

Based on Fig 7, it can be seen that the values are the same as those shown in Fig 6. The difference is clearly evident in the rate of decline. Using Equation (3), the rate of decline is calculated as follows:

$$Motor\ 1\ Drop = \frac{34.57-24.26}{34.57}100\% = 29.82\% \quad (7)$$

$$Motor\ 2\ Drop = \frac{34.57-24.96}{34.57}100\% = 27.8\% \quad (8)$$

Based on testing of the uncontrolled system using basketball throws, 3 out of 5 attempts successfully entered the hoop. Here, it can be observed that the uncontrolled system still exhibits poor performance. As seen in the calculations, the rise time and stabilization time are still too long, requiring a waiting period for the motor to stabilize before the ball can be thrown. In addition, both motors still exhibit overshoot; specifically, Motor 2 experienced a prolonged overshoot of 1.51% during the testing process. Similarly, the drop values when throwing the ball were as follows: Motor 1 = 29.82%, Motor 2 = 27.8%. The difference in drop values between the two motors was approximately 2%. Based on these uncontrolled tests, accuracy needs to be improved to achieve better and more optimal control.

B. System Testing with PID Control

In this test, PID values will be used to observe the motor's response when different values—P, PI, and PID—are applied. The first step in determining the P, PI, and PID values is to identify stable oscillations in the motor's response. To find this constant oscillation, the P value is gradually increased until a stable oscillation appears around the setpoint, as seen in the motor response curve, which forms peaks and troughs whose amplitude neither increases nor decreases, while the I and D values are set to 0. The stable oscillation referred to here is Kcr, or the critical gain. The following figure shows the oscillation that appears when determining Kcr.

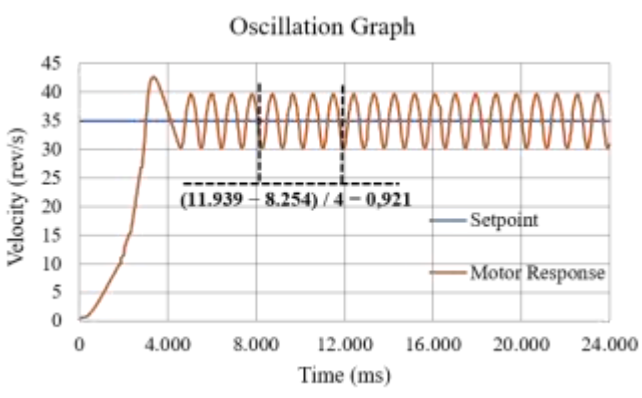

Fig 8. Oscillation on Kcr 160

P In Figure 8, the optimal oscillation occurs at a value of P = 160, which is then referred to as Kcr = 160. Meanwhile, this oscillation also has a Pcr, or critical period, with a value of 0.921. After obtaining the values of Kcr and Pcr, the P, PI, and PID constants can be calculated using the Ziegler-Nichols formula as follows.

TABLE IV ZIEGLER NICHOLS (ZN) PARAMETER VALUES

Controller	Kp	Ki	Kd
P	80	0	0
PI	72	0.7675	0
PID	96	0.4605	0.115125

In Table IV, the Ziegler-Nichols parameter values have yielded the P, PI, and PID values, and testing will then proceed for each controller. The first test was conducted using only the P controller. The following shows the motor response when using the P controller:

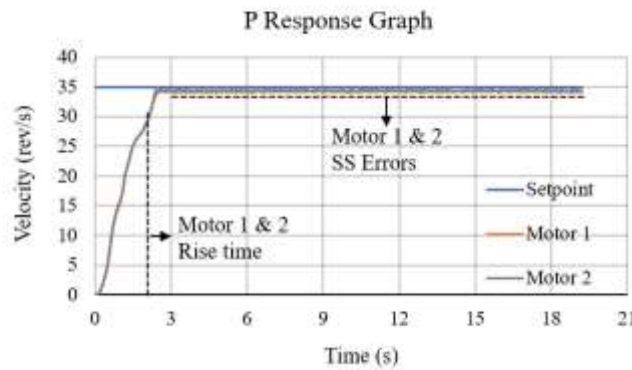

Fig 9. Response Graph for the P Controller

In Fig 9, the P-controller response shows that Motors 1 and 2 failed to reach the setpoint, so the settling time and overshoot were not observed. However, the rise time was still achieved within 1.8 seconds. The steady-state error can then be calculated using Equation (1). The following is the steady-state error for the P controller:

$$Error\ SS\ Motor\ 1\ \&\ 2 = \frac{35-34.11}{35}100\% = 2.54\% \quad (9)$$

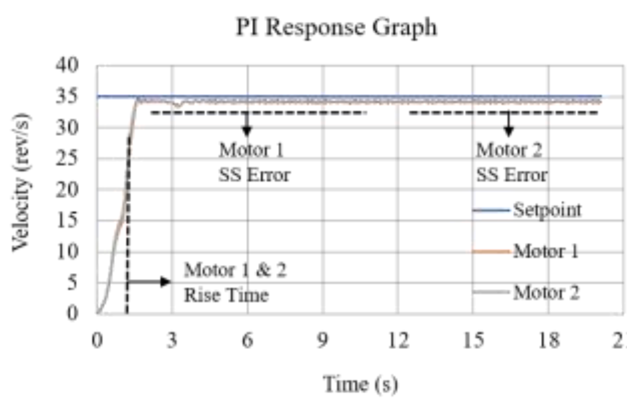

Fig 10. PI Controller Response Curve

In Figure 10, it can be seen that the PI controller response approaches the setpoint but does not quite reach it. There is no overshoot, but the rise time for both Motors 1 and 2 is 1.1 seconds. Similarly, the steady-state error can be calculated using Equation (1), which yields the following results:

$$Error\ SS\ Motor\ 1 = \frac{35-34.11}{35}100\% = 2.54\% \quad (10)$$

$$Error\ SS\ Motor\ 2 = \frac{35-34.14}{35}100\% = 2.45\% \quad (11)$$

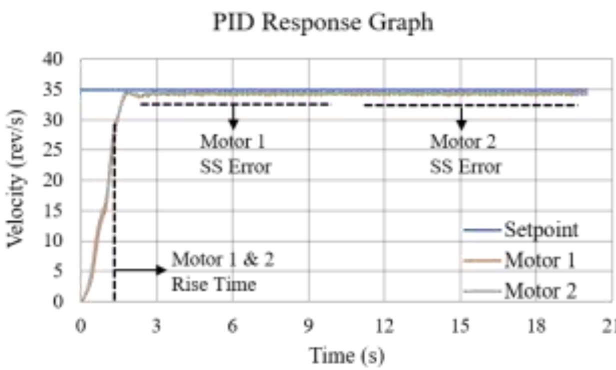

Fig 11. Response Graph for the PID Controller

In Fig 11, the response of the PID control system with Kp = 95, Ki = 0.4605, and Kd = 0.115125, as shown in TABLE IV, has not yet fully reached the setpoint; therefore, neither the settling time nor the overshoot is visible. However, the rise time for Motor 1 and Motor 2 is achieved within 1.2 seconds. The steady-state error can also be calculated using Equation (1) as follows:

$$Error\ SS\ Motor\ 1 = \frac{35-34.40}{35}100\% = 1.7\% \quad (12)$$

$$Error\ SS\ Motor\ 2 = \frac{35-34.36}{35}100\% = 1.8\% \quad (13)$$

Based on the P, PI, and PID tests using TABLE IV, the PID control performed better than the others, as evidenced by a motor rise time of 1.2 seconds, a steady-state error for motor 1 of 0.7%, and a steady-state error for motor 2 of 1.8%. However, these test results cannot yet be applied to shooting a basketball into the hoop because the system response has not fully reached the setpoint. Therefore, manual tuning is required using the previous PID values, and manual adjustments will be made to achieve better control.

C. System Testing Using Manual Adjustments

The previous PID controller analysis produced a fairly good response, but the settling time was still too long, meaning that to throw the ball, the user had to wait until the motor reached a stable state, this is clearly not ideal for effective and rapid motor control. To address this issue, manual tuning was required using the previous PID values [15]. Based on those values, manual tuning was performed to achieve better control, resulting in Kp = 100, Ki = 15, and Kd = 0.5. The response results can be seen in the following figure:

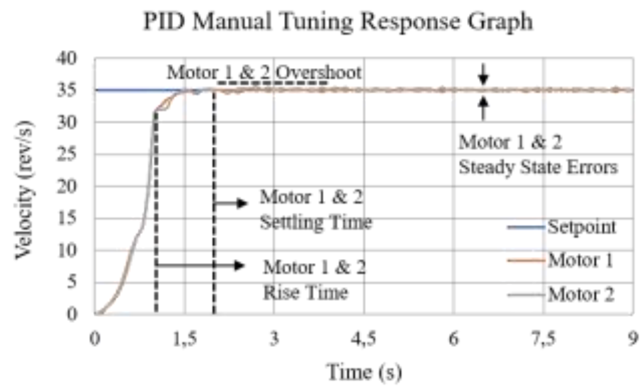

Fig 12. System response with manual PID tuning results

Fig 12 shows that the rise times for Motor 1 and Motor 2 are achieved within 1 second. Similarly, the settling times for Motor 1 and Motor 2 stabilize within 1.4 seconds. Using Equations (1) and (2), the overshoot and steady-state error values can be determined from the following calculations:

$$Error\ SS\ Motor\ 1 = \frac{35-34.99}{35}100\% = 0.005\% \quad (14)$$

$$Error\ SS\ Motor\ 2 = \frac{35-34.96}{35}100\% = 0.095\% \quad (15)$$

$$Overshoot\ Motor\ 1 = \frac{35.33-35}{35}100\% = 0.94\% \quad (16)$$

$$Overshoot\ Motor\ 2 = \frac{35.54-35}{35}100\% = 1.54\% \quad (17)$$

After obtaining the response values from the manually tuned PID controller, the next step is to conduct a test involving shooting a basketball into the hoop using the manually tuned PID values. The test consists of five trials, with a shooting distance of 2.5 meters and a hoop height of 2.43 meters, in accordance with the parameters shown in Fig 1.

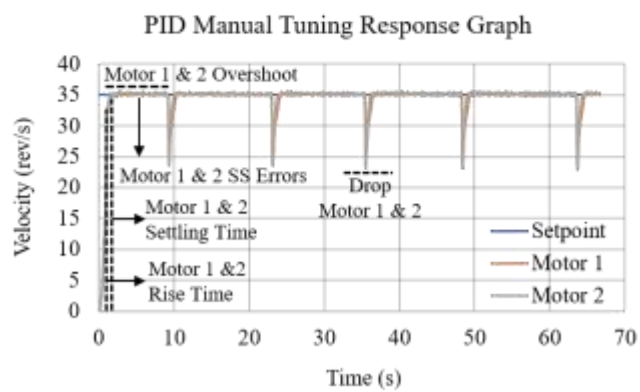

Fig 13. Response of the manually tuned PID system when throwing a ball

The results in Figure 13 show similarities to those in Figure 12. The difference lies in the drop value when throwing the ball toward the hoop. Using Equation (3), the drop value when throwing a basketball toward the hoop is as follows:

$$Drop\ Motor\ 1 = \frac{34.99-23.12}{34.99} 100\% = 33.26\% \quad (18)$$

$$Drop\ Motor\ 2 = \frac{34.96-23.22}{34.96} 100\% = 33.32\% \quad (19)$$

Based on the results of the rise time, settling time, overshoot, and steady-state error calculations, it can be seen that the manual tuning yielded better results, with 5 out of 5 attempts successfully landing in the ring. The rise time and settling time values are faster than those of other controllers, allowing the ball to be thrown into the ring earlier.

IV. Conclusion

Based on the test results aimed at achieving a good and stable ball throw—both without a controller and using the PID control method—it can be concluded that the uncontrolled system produces a suboptimal response, with a 60% success rate of the ball entering the hoop out of 5 trials, and is not recommended for applications requiring precision. The main problem with this uncontrolled system is the significant speed difference between Motor 1 and Motor 2, where Motor 1 is only able to approach the setpoint but not reach it, while Motor 2 tends to experience prolonged overshoot, albeit by only 1.51%. The rise time is still relatively long at 1.3 seconds; similarly, the settling time for Motor 1 is reached in 2.4 seconds and for Motor 2 in 2.1 seconds. The PID control system showed better results compared to the PI and PID controllers. However, the settling time was still relatively long, indicating the need for further manual tuning. After manual adjustments, improvements were observed with a rise time of 1 second and a settling time of 1.4 seconds. The steady-state error for Motor 1 was 0.005%, and for Motor 2, it was 0.095%. Similarly, the overshoot for both Motor 1 and Motor 2 was below 2%. After applying manual tuning, the response was noticeably improved, with a 100% success rate of the ball entering the ring out of 5 trials; therefore, it can be concluded that the manually tuned controller performs better than no control at all.

References

[1] P. Tryengineering, "Robot Basketball," *ABU Asia-Pacific Robot Contest 2025 Ulaanbaatar, Mong. Theme*, pp. 1–18, 2025, [Online]. Available: http://aburobocon2025.mnb.mn

[2] S. K. Krishnamoorthy, N. Das, P. Gudimetla, and K. Emami, "Enhanced Speed Control for BLDC Motors Using WOA-Integrated PID Controller Optimization," *IEEE Access*, vol. 12, pp. 162465–162475, 2024, doi: 10.1109/ACCESS.2024.3480349.

[3] F. Firdaus, E. Priatna, N. Hiron, and N. Busaeri, "Prototype Sistem Kendali Kecepatan Motor DC Dengan Proportional Integral Derivative (PID) Controller," Oktober.

[4] K. Mou, M. Zhang, D. Wang, M. Yang, and C. Liang, "Enhanced hippopotamus optimization algorithm for tuning proportional–integral–derivative controllers," *Front. Inf. Technol. Electron. Eng.*, vol. 26, no. 8, pp. 1–22, 2025, doi: 10.1631/FITEE.2400492.

[5] E. T. Maddalena, M. W. F. Specq, V. L. Wisniewski, and C. N. Jones, "Embedded PWM predictive control of DC-DC power converters via piecewise-affine neural networks," *IEEE Open J. Ind. Electron. Soc.*, vol. 2, no. January, pp. 1–8, 2021, doi: 10.1109/OJIES.2021.3058411.

[6] P. B. Gavali, S. . Patil, and S. M. Ingale, "Design and Fabrication of Thrower Mechanism for Multipurpose Ball Throwing Machine," *Asian Rev. Mech. Eng.*, vol. 11, no. 2, pp. 1–10, 2022, doi: /10.51983/arme-2022.11.2.3436.

[7] R. Damayanti, B. Syawaludin, and H. A. Gunawan, "Design of PI and PID Control System for Induction Motor Control on Gamma Irradiator Characterization Prototype Using a Modified Ziegler-Nichols Method," *J. Electr. Technol. UMY*, vol. 6, no. 2, pp. 1–9, 2022, doi: 10.18196/jet.v6i2.16412.

[8] Y. Hermanto and A. Kiswantono, "Stability Control of Frequency and Voltage in Wind Power Plant Using Complementary Load with Pid Control, Pwm and Thingspeak Monitor," *JEECS (Journal Electr. Eng. Comput. Sci.*, vol. 7, no. 1, pp. 1159–1168, 2023, doi: 10.54732/jeecs.v7i1.211.

[9] M. Huba, P. Bistak, J. Brieznik, and D. Vrancic, "Constrained Series PI, PID and PIDA Controller Design Inspired by Ziegler-Nichols," *Power Electron. Drives*, vol. 9, no. 1, pp. 1–16, 2024, doi: 10.2478/pead-2024-0021.

[10] C. R. Harahap, F. X. A. Setyawan, and C. A. Salsabila, "Iot Based Induction Motor Pid Speed Control System Using Ziegler – Nichols Method," *J. Eng. Sci. Technol.*, vol. 20, no. 4, pp. 1–20, 2025.

[11] L. Cavanini, F. Ferracuti, and A. Monteriù, "Optimal error governor for PID controllers," *Int. J. Syst. Sci.*, vol. 52, no. 12, pp. 2480–2492, 2021, doi: 10.1080/00207721.2021.1890272.

[12] M. O. Sihotang, T. Padang, R. P. Lubis, D. Jonto, and A. Sinuraya, "Analisis Respons Transien Motor DC terhadap Perubahan Beban Menggunakan Kontrol PI," vol. 3, no. 1, pp. 144–152, 2026.

[13] D. D. Anggraini, S. Palaloi, M. Diana, D. Maharani, and A. H. Sularso, "PID - Ziegler Nichols Method for Load Frequency Control," vol. 04, no. 02, pp. 63–67, 2025, doi: 10.20473/jatm.v4i2.69482.

[14] A. Maarif and N. R. Setiawan, "Control of dc motor using integral state feedback and comparison with pid: Simulation and arduino implementation," *J. Robot. Control*, vol. 2, no. 5, pp. 456–461, 2021, doi: 10.18196/jrc.25122.

[15] Z. Li, "Review of PID control design and tuning methods," *J. Phys. Conf. Ser.*, vol. 2649, no. 1, 2023, doi: 10.1088/1742-6596/2649/1/012009.