# Development of a Low-Level Control System Based on PID and RPM Imbalance Correction for Swarm Robotics

## Tugas Akhir


**Oleh:**
**Alexander Cang (4222201027)**

**Program Studi Teknik Robotika**
**Jurusan Teknik Elektro**
**Politeknik Negeri Batam**


**2025/2026**

# Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : “Development of a Low-Level Control System Based on PID and RPM Imbalance Correction for Swarm Robotics” adalah **hasil karya sendiri**, **diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan**, dan **bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.** Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 15 Januari 2026


Alexander Cang
NIM:  4222201027

# 1 Lembar Pengesahan

**Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar**
**Sarjana Terapan Teknik (S.Tr.T)**
**di**
**Politeknik Negeri Batam**

**Oleh:**
**Alexander Cang (4222201027)**

**Dengan judul:**
**Development of a Low-Level Control System Based on PID and RPM Imbalance Correction for Swarm Robotics**

**Tanggal Sidang: 15 01, 2026**

| Disetujui oleh : | |
|---|---|
| **Dosen Penguji I**<br><br>1. (Senanjung Prayoga, S.Pd., M.T)<br>NIP: 198908062019031015 | **Dosen Pembimbing I**<br><br>1. Anugerah Wibisana, S.Tr.T., M.Tr.T<br>NIP: 199404272024061001 |
| **Dosen Penguji II**<br><br>2. (Rifky Afriza, S.Tr.T., M.Sc)<br>NIP: 199404062025061003 | |

# Development of a Low-Level Control System Based on PID and RPM Imbalance Correction for Swarm Robotics

**Alexander Cang[1], Anugerah Wibisana[1]**
[1]Department of Electrical Engineering, Robotics Engineering Technology Study Program, Politeknik Negeri Batam, Batam, Indonesia

*Email: Alexandercang18@gmail.com

*Received on dd-mm-yyyy* | *Revised on dd-mm-yyyy* | *Accepted on dd-mm-yyyy*

***Abstract*— Swarm robots are robotic systems capable of coordinated movement in formation patterns, with roles divided into leader and follower robots. Motor speed imbalance and trajectory deviations in differential drive mobile robots can lead to position and orientation errors, negatively affecting the accuracy of target motion. This study aims to design a motor speed control system that combines a Proportional–Integral–Derivative (PID) controller with an RPM imbalance correction method, along with odometry-based evaluation of individually controlled robots. Experimental results from motor characterization using encoders indicate motor speed variations below 5%. Testing three PID parameter configurations identified an optimal setting capable of maintaining stable motor speed, while the RPM imbalance correction method successfully maintained motor speed within the range of 49–51 RPM relative to the 50 RPM target, with errors below 1%, reduced overshoot, and faster settling time. Odometry-based evaluation during triangular and square formation tests showed average distance and orientation errors below 2%, demonstrating that the proposed control system effectively improves motion stability and accuracy.**

**Keywords: Swarm robot, Differential drive, PID control, RPM imbalance Correction, Odometry, Formation motion.**

## I. INTRODUCTION

Swarm robotics refers to systems of multiple robots that move and coordinate cooperatively, typically using a leader–follower formation. This approach enables more efficient task execution compared to single-robot systems [1]. In differential-drive swarm robots, motion instability often occurs due to speed discrepancies between left and right wheel motors, causing trajectory deviations and non-linear movement [2, 3]. Such imbalances directly impact navigation accuracy and formation stability, particularly in maintaining consistent inter-robot distance and orientation.

To address these challenges, this study implements a low-level motor speed control system combining PID control and RPM imbalance correction to maintain wheel speed equilibrium. PID is used for its simple structure and stable response in DC motor speed control [4], while RPM imbalance correction adaptively reduces speed differences between wheels [5, 6]. Wheel characteristic testing is conducted to inform control adjustments, and odometry is employed to quantitatively evaluate target distance and orientation in formation experiments [7, 8]. Unlike most previous studies that focus on motor speed control in isolation [9, 10], this study integrates wheel testing, RPM imbalance correction, and odometry evaluation to ensure uniform motion responses, minimize cumulative errors, and maintain formation stability [2, 3, 11].

## II. METHOD

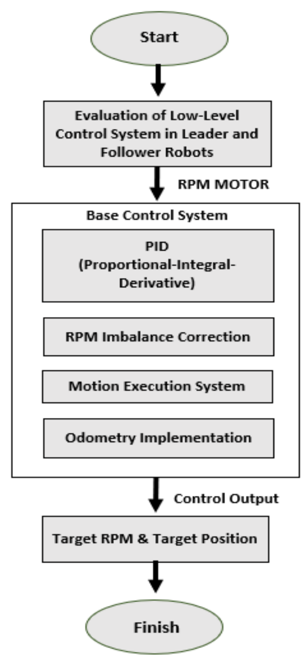



Figure 1. Flowchart of the Low-Level Control System for Robot Leader and Follower

Figure 1 shows the research flowchart illustrating the main stages of this study. It begins with the design and evaluation of a low-level control system based on PID and RPM imbalance correction, followed by its implementation on leader and follower robots. Robot motion testing is then conducted to assess target RPM tracking and position accuracy in simple formation configurations. Finally, the experimental data are analyzed to evaluate the performance of the proposed control system.

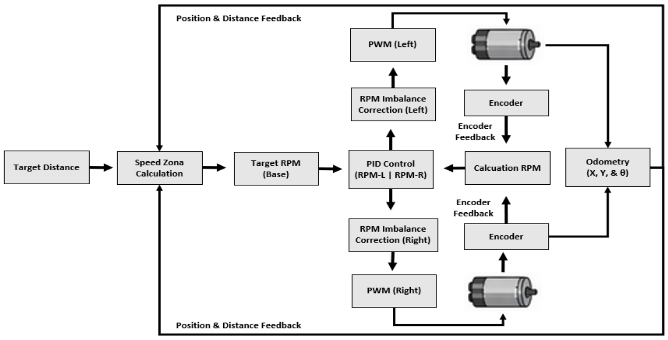



Figure 2. Block Diagram of the Low-Level Control and Odometry System Implemented on The Robot.

Figure 2 illustrates the architecture of the low-level motor control system. The speed zone calculation module receives the robot's traveled distance obtained from the forward odometry system and compares it with the target distance to determine the current motion phase. Based on this process, a base target RPM is generated as the reference speed. This reference is processed by the PID controllers of the left and right motors to produce control signals for the motor driver, followed by an RPM imbalance correction mechanism to compensate for speed differences between both motors, ensuring smoother and more stable motion. The actual motor speeds are measured using encoders and fed back into the system to form a closed-loop control structure. Encoder data are also utilized by the odometry system to estimate the robot's position and orientation for motion performance evaluation.

### A. Evaluation of Low-Level Control in Leader–Follower Robots

The low-level control of leader–follower robots aim to ensure stable and balanced DC motor speed tracking of the target RPM. N20 DC motors are used, which naturally exhibit variations due to differences in winding properties, gear ratios, and installation positions, potentially causing speed discrepancies between left and right wheels. A PID controller is implemented to minimize the error between target and actual RPM, providing an effective method for stable motor speed control [4]. Motor characteristics are tested by applying PWM signals across their full range to compare left and right motor responses, and the system's performance is evaluated using encoder feedback to assess RPM tracking accuracy.

### B. PID (Proportional-Integral-Derivative)

In low-level control system, a PID (Proportional–Integral–Derivative) controller is used to regulate the speed of the DC motor. The PID receives the error between the target RPM and the actual RPM from the encoder and generates a PWM signal to precisely control the motor [12]. Mathematically, the PID control law is expressed as:

$$u(t) = K_{pe(t)} + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \qquad (1)$$

In (1), $u(t)$ denotes the PWM control signal, $e(t)$ is the speed error between the target and actual RPM, and $K_p$, $K_i$, and $K_d$ are the proportional, integral, and derivative gains of the PID controller.

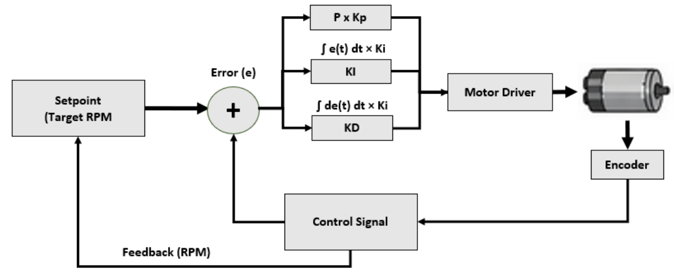



Figure 3. PID Control System for Motor DC

Figure 3 illustrates the PID block diagram, where the error is processed through the P, I, and D paths and combined into a PWM signal for the motor driver, with encoder feedback enabling real-time RPM tracking. Each PID parameter serves a specific role: $K_P$ accelerates response, $K_I$ eliminates steady-state error, and $K_D$ dampens rapid changes to improve stability and reduce overshoot [13]. Parameters are tuned experimentally through trial and error, which is effective for DC motors with nonlinear characteristics.

### C. RPM Imbalance Correction

RPM imbalance correction balances the speed of the left and right motors to ensure smoother and more stable robot motion, compensating for mechanical and electrical differences that cause speed discrepancies [14]. The method calculates the speed difference ($\Delta RPM = RPM_{Left} - RPM_{Right}$) from encoder feedback and generates a correction factor to adjust the target RPMs of both motors.

$$RPM_{targetleft} = RPM_{base} - K_f \cdot \Delta RPM \qquad (2)$$
$$RPM_{targetright} = RPM_{base} + K_f \cdot \Delta RPM \qquad (3)$$

RPM imbalance correction balances the speed of the left and right motors to ensure smoother and more stable robot motion, compensating for mechanical and electrical differences that cause speed discrepancies [14]. The method calculates the speed difference as $\Delta RPM = RPM_{Left} - RPM_{Right}$ using encoder feedback and generates a correction factor to adjust the target RPMs of both motors. In (2), $RPM_{target,left}$ and $RPM_{target,right}$ represent the adjusted target speeds for the left and right motors, based on the base speed ($RPM_{base}$), correction gain ($K_f$), and the measured speed difference ($\Delta RPM$). Acting as an auxiliary mechanism, RPM imbalance correction provides a simple, computationally light, and easily implementable approach that adaptively balances motor speeds and effectively improves robot movement [14, 15].

### D. Motion Execution System

The motion execution system serves as the main decision-making layer for driving the robot to reach specified distance and orientation targets. It uses a rule-based approach with conditional logic (if–else) and a finite state machine (FSM), dividing robot movement into states such as forward motion (MOVE), rotation (TURN), and final (FINISH). State transitions are controlled using a switch(state) structure, with decisions based on wheel encoder data. The robot remains in the forward state until the target distance is reached, then transitions to rotation, continuing until the angular error falls within tolerance. This deterministic approach ensures that both linear and rotational motions are accurately executed based on real-time encoder feedback.

In the forward motion state, the distance traveled by each wheel is calculated from the change in encoder pulses as follows:

$$\Delta s_L = \frac{2\pi R \ \cdot \Delta N_L}{PPR \ . \ G}, \qquad \Delta s_R = \frac{2\pi R \ \cdot \Delta N_R}{PPR \ . \ G} \qquad (4)$$

In (4), $\Delta s_L$ and $\Delta s_R$ represent the linear distances traveled by the left and right wheels, respectively. These distances are calculated from the change in encoder pulses ($\Delta N_L$ *and* $\Delta N_R$) scaled by the wheel circumference $2\pi R$ ($2\pi R$) and divided by the product of pulses per revolution ($PPR$) and gear ratio ($G$).

*E. Odometry Implementation*

Odometry estimates a mobile robot's position and orientation using wheel movement data [3]. In this study, it provides position coordinates (x, y) and orientation (θ) for evaluating robot motion along formation paths and assessing target position accuracy. The robot employs a differential-drive configuration with two encoder-equipped wheels, generating pulses proportional to wheel rotation [7].

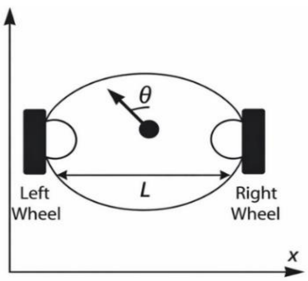



Figure 4. Kinematic Model Of The Differential Drive Robot

Figure 4 illustrates the kinematic model of the differential-drive robot used in this study, showing the left and right wheels, wheelbase LLL, and the coordinate system (x, y) with orientation θ. This provides a clear reference for the subsequent mathematical odometry calculations. The distance traveled by each wheel is calculated as follows:

$$\Delta s_L = \frac{2\pi R \ \cdot \Delta N_L}{N_{enc}}, \qquad \Delta s_R = \frac{2\pi R \ \cdot \Delta N_R}{N_{enc}} \qquad (5)$$

In (5), $\Delta s_L$ and $\Delta s_R$ represent the linear distances traveled by the left and right wheels, calculated from encoder pulse changes ($\Delta N_L$ and $\Delta N_R$), scaled by the wheel circumference ($2\pi R$) and divided by the pulses per revolution ($N_{enc}$). This converts encoder pulses into actual linear distances for forward motion control and orientation estimation. The robot's overall linear displacement ($\Delta_s$) and orientation change ($\Delta_\theta$) are then calculated as follows:

$$\Delta s \ = \frac{\Delta s_R + \ \Delta s_L}{2}, \qquad \Delta\theta \ = \frac{\Delta s_R - \ \Delta s_L}{L} \qquad (6)$$

In (6), $\Delta s$ represents the robot's overall linear displacement, calculated as the average of the left and right wheel distances ($\Delta s_L$ and $\Delta s_R$). $\Delta\theta$ represents the change in the robot's orientation, calculated from the difference between the right and left wheel distances divided by the wheelbase ($L$). This calculation converts the individual wheel movements into the robot's forward displacement and rotational change, which are used to update its position and heading.

$$x_{k+1} = x_k + \Delta s \cos(\theta_k) \qquad (7)$$

$$y_{k+1} = y_k + \Delta s \sin(\theta_k) \qquad (8)$$

$$\theta_{k+1} = \theta_k + \Delta\theta \qquad (9)$$

In (7), $x_{k+1}$ and $y_{k+1}$ are the updated robot coordinates in the global frame, calculated by projecting the linear displacement $\Delta s$ along the current heading $\theta_k$ using cosine and sine functions. $\theta_{k+1}$ is the updated orientation, obtained by adding $\Delta\theta$ to $\theta_k$. This method converts wheel movements into global position and orientation updates, which are essential for odometry-based motion tracking. Odometry calibration is sometimes applied to reduce wheel slip and mechanical inaccuracies [10].

*F. Target RPM & Target Position*

Target RPM and robot positions were tested using formation-based trials, where each formation demonstrated the robots' ability to move smoothly and stably. The movement results were compared with the predetermined targets within the formation scheme, including the robots' traveled distance, to evaluate the extent to which the robots could accurately achieve the target RPM and target positions.

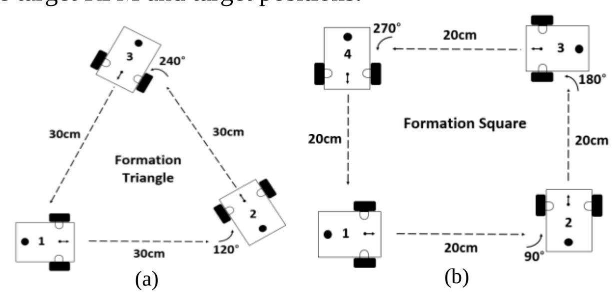



Figure 5. Formation Scheme of The Robot (a) Triangle & (b) Rectangle

Figure 5 illustrates the robot formation testing design, where the robot moves forward until it reaches the predetermined distance, then rotates according to the targeted orientation ($\theta$). This forward motion and rotation process is repeated until the desired formation, such as a triangle or square, is fully formed. The odometry system is used to calculate the robot's linear displacement and orientation change based on encoder pulse data from the left and right wheels, allowing position ($x$, $y$) and orientation ($\theta$) information to manage state transitions and ensure the robot follows the target formation accurately.

Odometry is used in the tests to evaluate the performance of the PID controller and RPM imbalance correction, ensuring smooth motion and accurate tracking of predefined distance and orientation targets. Motor speed is controlled using PID combined with RPM imbalance correction to maintain stability and alignment between the left and right wheels.

## III. RESULT AND DISCUSSION

### A. Motor RPM Measurements under PWM Variations

Measured RPM values of the left and right motors, obtained from serial data readings, reveal a speed difference between the two motors, particularly at low to medium PWM levels, reflecting the non-identical characteristics of each motor. This discrepancy can affect the stability and accuracy of the robot's motion. These RPM measurements serve as the baseline before implementing the PID controller. Table I presents the left and right motor RPMs for each robot along with their comparison in percentage form, where the RPM imbalance percentage is calculated as the difference between the motors divided by their average RPM. This provides a quantitative representation of motor speed imbalance, serving as a reference for PID tuning and for evaluating performance improvements after applying the controller.

TABLE I
Measured RPM of Left and Right Motors at Leader and Follower Robots

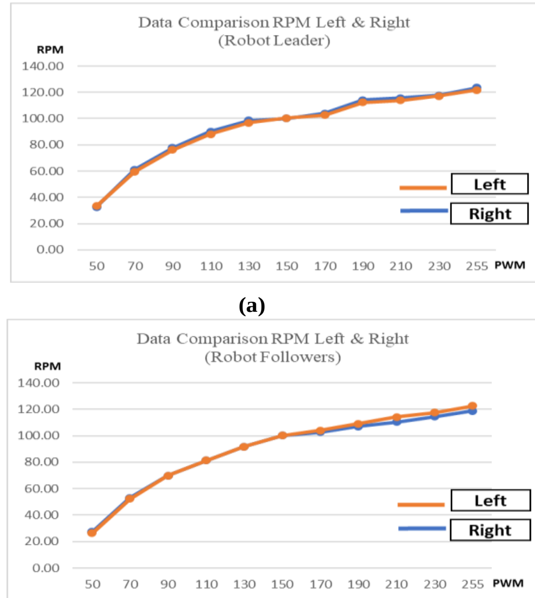
| PWM | Robot Leader | | | Robot Follower | | |
|---|---|---|---|---|---|---|
| | RPM_L | RPM_R | Diff. (%) | RPM_L | RPM_R | Diff. (%) |
| 50 | 33.43 | 32.57 | 2.61 | 26.40 | 27.43 | 3.83 |
| 70 | 59.66 | 61.20 | 2.55 | 52.29 | 52.97 | 1.29 |
| 90 | 76.29 | 77.83 | 2.00 | 69.77 | 69.77 | 0.00 |
| 110 | 88.11 | 90.17 | 2.31 | 81.26 | 81.26 | 0.00 |
| 130 | 96.69 | 98.57 | 1.93 | 91.71 | 91.71 | 0.00 |
| 150 | 100.31 | 100.11 | 0.20 | 100.11 | 100.11 | 0.00 |
| 170 | 102.69 | 104.06 | 1.33 | 103.89 | 102.69 | 1.16 |
| 190 | 112.46 | 114.00 | 1.36 | 109.03 | 107.14 | 1.75 |
| 210 | 113.83 | 115.54 | 1.49 | 114.17 | 110.40 | 3.36 |
| 230 | 117.09 | 117.77 | 0.59 | 117.60 | 114.34 | 2.81 |
| 255 | 121.89 | 123.43 | 1.26 | 122.57 | 118.63 | 3.27 |

Table 1, it can be observed that the average error for each robot is below 5%. This phenomenon is commonly encountered when using DC motors, as each motor has mechanical and electrical characteristics that are not entirely identical. These differences include coil resistance, gearbox condition and friction, as well as encoder resolution and reading noise, which vary between individual motors.

Differences in characteristics between the left and right motors become more apparent in Figure 6, where the graph shows discrepancies in the robot's movement trajectory as the PWM values increase. At higher PWM levels, the motors operate closer to their maximum speed, causing the mechanical motion within the gearbox to become faster and more complex. This condition makes the dynamic characteristics of each motor's gearbox more dominant, ultimately affecting the uniformity of speed between the left and right motors.

### B. Tunning Parameter PID

The PID controller was tested to regulate motor speed to a target of 50 RPM, chosen as it falls within the low-to-medium range and is sensitive to parameter changes. Three different combinations of $K_p$, $K_i$, and $K_d$ were applied to both leader and follower robots, with performance evaluated based on overshoot, settling time, steady-state error, and consistency




**(a)**

**(b)**

Figure 6. Comparison Graph of Left and Right Motor RPM of the Robot (a) Leader & (b) Follower

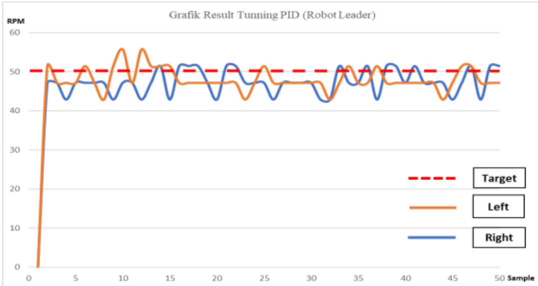
TABLE II
PID Parameters for Leader and Follower Robots

| Robot | Kp R | Kp L | Ki R | Ki L | Kd R | Kd L |
|---|---|---|---|---|---|---|
| Leader Test 1 | 3.35 | 3.17 | 0.038 | 0.032 | 0.05 | 0.43 |
| Follower Test 1 | 2.78 | 2.75 | 0.06 | 0.055 | 0.075 | 0.066 |
| Leader Test 2 | 4.50 | 4.13 | 0.05 | 0.035 | 0.06 | 0.05 |
| Follower Test 2 | 2.85 | 2.75 | 0.06 | 0.055 | 0.07 | 0.066 |
| Leader Test 3 | 4.90 | 4.65 | 0.21 | 0.206 | 0.02 | 0.02 |
| Follower Test 3 | 3.9 | 3.55 | 0.06 | 0.06 | 0.05 | 0.05 |

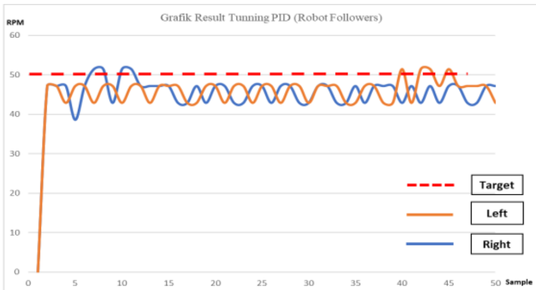
TABLE III
Performance Results of the PID Controller for Leader and Follower Robots.

| Robot | Overshoot (%) | Settling time (s) | Steady-state error (RPM) | Explanation |
|---|---|---|---|---|
| Leader Test 1 | 11.42 | 0.25 | 0.72 | Unstable |
| Follower Test 1 | 2.86 | 0.10 | 0.72 | |
| Leader Test 2 | 2.86 | 0.03 | 0.72 | Undeshoot |
| Follower Test 2 | 2.86 | 0.04 | 0.72 | |
| Leader Test 3 | 11.42 | 1.00 | 0.72 | Optimal |
| Follower Test 3 | 11.42 | 1.00 | 0.72 | |

Figure 7, the initial tuning results show an unstable system response, characterized by the presence of undershoot and overshoot. The average speed reaches only about 42 RPM, which is lower than the target of 50 RPM, while the overshoot is recorded at approximately 55 RPM. This condition is caused by a proportional gain ($K_P$) that is too low and integral ($K_I$) and derivative ($K_D$) parameters that are not yet optimal, resulting in the system's inability to achieve stable performance. Furthermore, based on tests conducted on 50 data samples, the



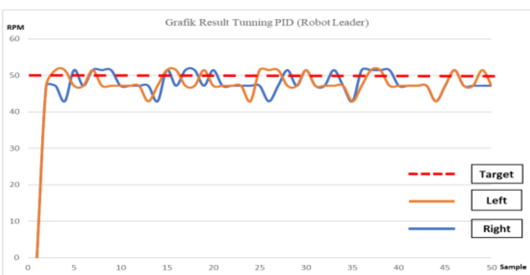

**(a)**



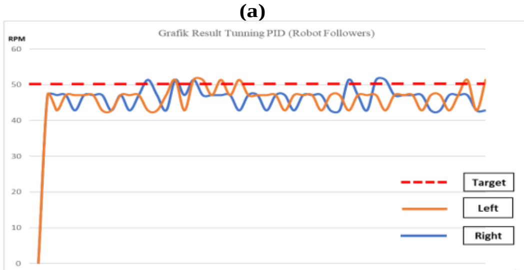

**(b)**

Figure 7. PID Response Graph with Initial Parameters of the Robot (a) Leader & (b) Follower

Figure 8, the results second tuning result shows a more stable response without overshoot; however, undershoot is still observed, with the motor speed reaching only about 42 RPM compared to the 50 RPM target. This condition is caused by a relatively low $K_P$ value and non-optimal $K_I$ and $K_D$ parameters



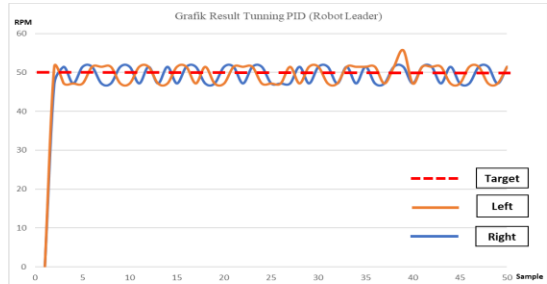

**(a)**



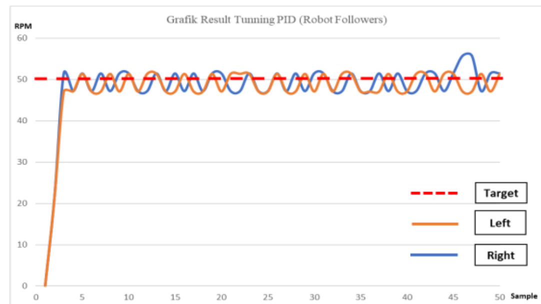

**(b)**

Figure 8. PID Response Graph with Second Tuning Parameters of the Robot (a) Leader & (b) Follower

Figures 9, The result third tuning demonstrates the best controller performance compared to the previous tuning results, with a more stable system response and motor speed reaching close to the target range of 47–51 RPM. Based on 50 data samples, the resulting RPM fluctuations are smaller and more consistent, indicating improved speed stability. Although a slight overshoot is still observed, it is considered acceptable and mainly caused by encoder characteristics and measurement noise commonly found in DC motor systems. Overall, the system is able maintain motor speed consistently without significant oscillations, confirming that the implemented PID controller operates optimally.




**(a)**




**(b)**

Figure 9. PID Response Graph with Optimal PID Parameters of the Robot (a) Leader & (b) Follower

### *C. RPM Imbalance Correction*

RPM imbalance correction is applied as an additional correction mechanism to the PID controller output to reduce speed differences between the left and right motors. The correction adjusts the control signals based on the measured RPM difference, resulting in shorter transient response, reduced overshoot, and improved speed uniformity, thereby enabling more stable, accurate, and precise straight-line motion of the robot.

Table IV, the RPM imbalance correction logic is designed to equalize the RPM differences between the left and right motors. If the RPM difference is ≤ 3, the controller does not apply any correction. When the difference is between 3 and 8 RPM, the controller applies a correction of 1 RPM by increasing or decreasing the target speed of one motor to reduce the imbalance. The larger the RPM difference, the greater the correction applied, allowing the system to adjust the speed discrepancy proportionally.

TABLE IV
Logic of the RPM Imbalance Correction Controller Robots.

| No | Speed Difference (ΔRPM) | Correction Value (ΔRPM) | Explanation |
|---|---|---|---|
| 1 | ≤ 3 RPM | 0 | No Correction |
| 2 | > 3 – 8 RPM | 1 | Small Correction |
| 3 | > 8 – 15 RPM | 2.5 | Medium Correction |
| 4 | > 15 RPM | 4 | Hight Correction |

Tables V and VI show that during the initial motion, the encoder records low RPM values as the system response. As the speed approaches the target, the RPM imbalance correction works together with the PID controller by compensating for the speed difference between the left and right motors. Experimental results indicate that this mechanism effectively reduces overshoot and undershoot, leading to a more stable RPM response. The tables present the percentage RPM deviation before and after correction, calculated from the difference between both motor speeds relative to their average. The results show that the average RPM deviation, initially ranging from 1.5% to 2.8% using PID control alone, was reduced to below 1% after applying RPM imbalance correction, demonstrating the effectiveness of the proposed method.

TABLE V
Comparison of left and right motor RPM before and after RPM imbalance correction implementation at leader

| NO | Robot Leader | | | | | |
|---|---|---|---|---|---|---|
| | PID RPM L | PID RPM R | Dev. (PID) (%) | Corr. RPM L | Corr. RPM R | Dev. (Corr.) (%) |
| 1 | 21.43 | 21.43 | 57.1 | 47.1 | 47.1 | 5.8 |
| 2 | 42.90 | 51.40 | 5.7 | 52.50 | 47.50 | 0.0 |
| 3 | 51.40 | 51.40 | 2.8 | 50.00 | 50.00 | 0.0 |
| 4 | 55.70 | 47.10 | 2.8 | 47.50 | 52.50 | 0.0 |
| 5 | 51.40 | 51.40 | 2.8 | 50.00 | 50.00 | 0.0 |
| 6 | 47.10 | 51.40 | 1.5 | 51.00 | 49.00 | 0.0 |
| 7 | 51.40 | 47.10 | 1.5 | 49.00 | 51.00 | 0.0 |
| 8 | 51.40 | 47.10 | 1.5 | 49.00 | 51.00 | 0.0 |
| 9 | 47.10 | 51.40 | 1.5 | 51.00 | 49.00 | 0.0 |
| 10 | 47.10 | 51.40 | 1.5 | 51.00 | 49.00 | 0.0 |

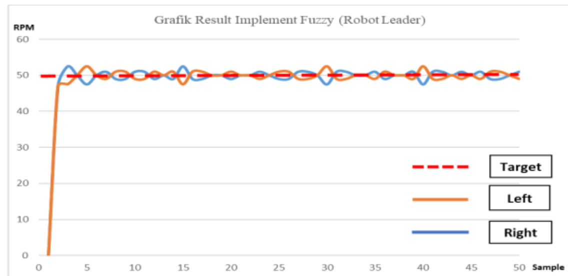
TABLE IV
Comparison of left and right motor RPM before and after RPM imbalance correction implementation at Follower

| NO | Robot Follower | | | | | |
|---|---|---|---|---|---|---|
| | PID RPM L | PID RPM R | Dev. (PID) (%) | Corr. RPM L | Corr. RPM R | Dev. (Corr.) (%) |
| 1 | 21.43 | 21.43 | 57.1 | 47.1 | 47.1 | 5.8 |
| 2 | 47.1 | 51.40 | 1.5 | 51.00 | 49.00 | 0.0 |
| 3 | 47.1 | 47.10 | 5.8 | 50.00 | 50.00 | 0.0 |
| 4 | 51.4 | 47.10 | 1.5 | 49.00 | 51.00 | 0.0 |
| 5 | 47.1 | 47.10 | 5.8 | 50.00 | 50.00 | 0.0 |
| 6 | 47.1 | 47.10 | 5.8 | 50.00 | 50.00 | 0.0 |
| 7 | 47.1 | 47.10 | 5.8 | 50.00 | 50.00 | 0.0 |
| 8 | 47.1 | 51.40 | 1.5 | 51.00 | 49.00 | 0.0 |
| 9 | 47.1 | 51.40 | 1.5 | 51.00 | 49.00 | 0.0 |
| 10 | 47.1 | 51.40 | 1.5 | 51.00 | 49.00 | 0.0 |

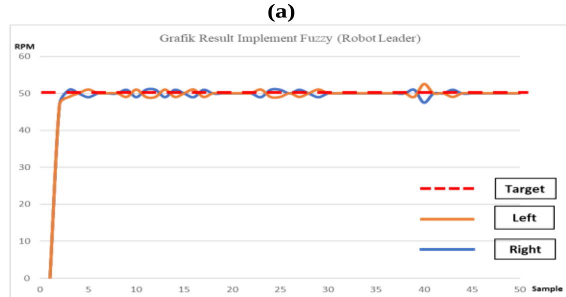
As shown in Figure 10, the graph illustrates that the PID response is successfully corrected by RPM imbalance correction**.** RPM imbalance correction effectively reduces undershoot and overshoot, keeps the system more stable at the target position, and shortens the settling time, resulting in RPM values without significant oscillations.

*D. Distance and Orientation Measurement*

Based on the test results, the odometry system is able provide information on the robot's traveled distance and orientation change (θ) obtained from the left and right wheel encoder data. This information is used to analyze the robot's movement performance in supporting the formation creation and maintenance. The odometry test results are presented refer to Tables VII and VIII, which the robot's traveled distance and orientation changes during the tests.




**(a)**




**(b)**

Figure 10. Results RPM Imbalance Correction for Robot (a) Leader and (b) Follower

TABLE VII
Results of Robot Travel Distance Testing (Target vs Actual)

| NO | Tar. Dis. (cm) | Dis. Tol. (cm) | Robot Leader | | | Robot Follower | | |
|---|---|---|---|---|---|---|---|---|
| | | | Act. Dis. (cm) | Dis. Dev. (cm) | Dis. (%) | Act. Dis. (cm) | Dis. Dev. (cm) | Dis. (%) |
| 1 | 10 | 0.2 | 9.8 | 0.2 | 0.020 | 9.8 | 0.2 | 0.020 |
| 2 | 15 | 0.2 | 14.8 | 0.2 | 0.013 | 14.8 | 0.2 | 0.013 |
| 3 | 20 | 0.2 | 19.8 | 0.2 | 0.010 | 19.8 | 0.2 | 0.010 |
| 4 | 25 | 0.2 | 24.8 | 0.2 | 0.008 | 24.8 | 0.2 | 0.008 |
| 5 | 30 | 0.2 | 29.8 | 0.2 | 0.007 | 29.8 | 0.2 | 0.007 |
| 6 | 35 | 0.2 | 34.8 | 0.2 | 0.006 | 34.8 | 0.2 | 0.006 |
| 7 | 40 | 0.2 | 39.8 | 0.2 | 0.005 | 39.8 | 0.2 | 0.005 |
| 8 | 45 | 0.2 | 44.8 | 0.2 | 0.004 | 44.8 | 0.2 | 0.004 |
| 9 | 50 | 0.2 | 49.8 | 0.2 | 0.004 | 49.8 | 0.2 | 0.004 |
| 10 | 55 | 0.2 | 54.8 | 0.2 | 0.004 | 54.8 | 0.2 | 0.004 |

Based on the results in Table VII, both the leader and follower robots achieved travel distances close to the target values in all test scenarios. The differences between the target and measured distances were small, with the maximum error remaining within the ±0.2 cm tolerance. The table presents distance deviations in both centimeters and percentage form, calculated from the difference between the target and measured distances, to facilitate analysis in absolute and relative terms. Overall, the odometry system demonstrated consistent and accurate distance measurement, with percentage errors below 2%, indicating stable and reliable robot movement throughout the tests.

Orientation test results refer to Table VIII show that the actual angles achieved by both the leader and follower robots were close to the target angles. The errors recorded in all tests remained within the tolerance limit of ±2°. Additionally, the table presents both the angular differences in degrees and the percentage errors, which are calculated based on the difference between the target and measured angles. Presenting both values aim to facilitate analysis and understanding of the deviations in robot orientation, both in absolute and relative terms. This indicates that the robot orientation control system is capable maintaining the heading accurately. The minor differences observed are likely due to mechanical and sensor factors but do not significantly affect the overall system performance.

TABLE VIII
Results of Robot Orientation Testing (Target vs Actual)

| NO | Targe (DEG°) | Tol. (DEG°) | Robot Leader | | | Robot Follower | | |
|---|---|---|---|---|---|---|---|---|
| | | | Act. Dis. (cm) | Dis. Dev. (cm) | Dis. (%) | Act. Dis. (cm) | Dis. Dev. (cm) | Dis. (%) |
| 1 | 360 | 2 | 361.0 | 1.0 | 0.00 | 358.9 | 1.1 | 0.00 |
| 2 | 45 | 2 | 44.5 | 0.5 | 0.01 | 43.3 | 1.7 | 0.03 |
| 3 | 90 | 2 | 90.2 | 0.2 | 0.00 | 89.0 | 1.0 | 0.01 |
| 4 | 135 | 2 | 133.1 | 1.9 | 0.01 | 134.3 | 0.7 | 0.00 |
| 5 | 180 | 2 | 179.5 | 0.5 | 0.00 | 181.4 | 1.4 | 0.00 |
| 6 | 225 | 2 | 223.6 | 1.4 | 0.00 | 223.8 | 1.2 | 0.00 |
| 7 | 270 | 2 | 270.3 | 0.3 | 0.00 | 271.5 | 1.5 | 0.00 |
| 8 | 315 | 2 | 314.4 | 0.6 | 0.00 | 313.1 | 1.9 | 0.00 |

### *E. Formation Motion*

Results of the robot formation tests are presented based on the distances between robots and the predetermined orientation angles (θ). These tests were conducted to evaluate the robots' ability to form and maintain formations using the PID controller, RPM imbalance correction method, and odometry system.

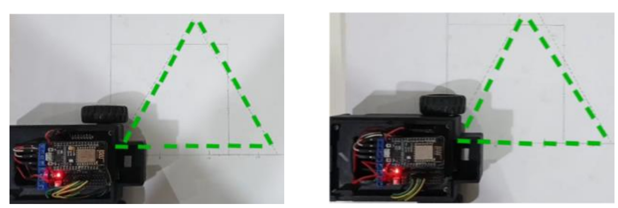
Based on the test results shown in Table IX and X, the robot formation performance demonstrates good capability in achieving the target distances. The distances achieved remained within the specified tolerance limits, with relative errors remaining small and below 1%. Additionally, Table 7 presents the percentage distance differences (distance %) calculated based on the difference between the target and actual distances. Presenting this value facilitates the analysis of robot movement deviations in relative terms. This indicates that the control system is capable maintaining accurate motion for both the leader and follower robots during the formation process.

TABLE IX
Results of Distance Testing at Robot Formation Experiments (Distance)

| Robot | Formation | Target Distance (cm) | Actual Distance (cm) | Distance (%) |
|---|---|---|---|---|
| Leader Test 1 | Triangle | 15 | 16 | 0.07 |
| Follower Test 1 | Triangle | 15 | 16 | 0.07 |
| Leader Test 2 | Triangle | 30 | 31.5 | 0.05 |
| Follower Test 2 | Triangle | 30 | 31 | 0.03 |
| Leader Test 3 | Square | 10 | 11 | 0.10 |
| Follower Test 3 | Square | 10 | 11 | 0.10 |
| Leader Test 4 | Square | 20 | 21 | 0.05 |
| Follower Test 4 | Square | 20 | 21.5 | 0.08 |

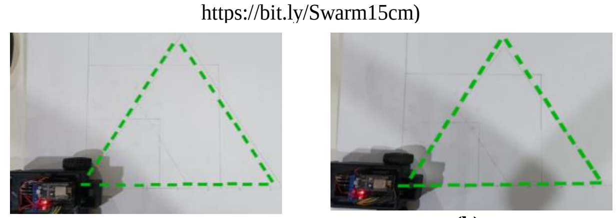
TABLE X
Results of Distance Testing at Robot Formation Experiments (Angle)

| Robot | Formation | Target Angle (DEG°) | Actual Angle (DEG°) | Distance (%) |
|---|---|---|---|---|
| Leader Test 1 | Triangle | [120, 240, 360] | [120, 242, 360] | 0.42 |
| Follower Test 1 | Triangle | [120, 240, 360] | [120, 242, 360] | 0.42 |
| Leader Test 2 | Triangle | [120, 240, 360] | [120, 242, 360] | 0.42 |
| Follower Test 2 | Triangle | [120, 240, 360] | [120, 241, 360] | 0.42 |
| Leader Test 3 | Square | [90, 180, 270, 360] | [91, 180, 271, 360] | 0.49 |
| Follower Test 3 | Square | [90, 180, 270, 360] | [90, 181, 271, 363] | 0.31 |
| Leader Test 4 | Square | [90, 180, 270, 360] | [90, 182, 272, 360] | 0.62 |
| Follower Test 4 | Square | [90, 180, 270, 360] | [90, 180, 271, 360] | 0.12 |



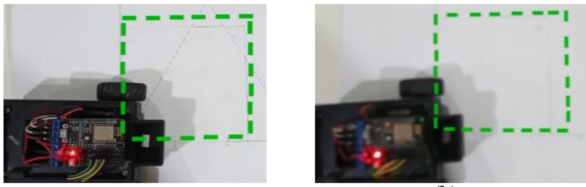
**(a)** **(b)**

Figure 11. Triangle formation test visualization with 15 cm spacing at robot (a) leader and (b) follower (Video demonstration available at: https://bit.ly/Swarm15cm)



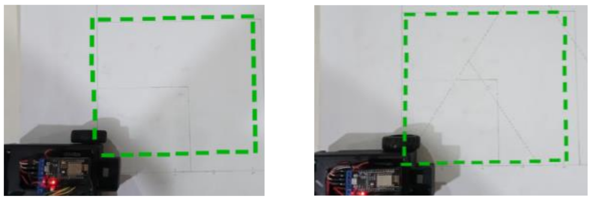
**(a)** **(b)**

Figure 12. Triangle formation test visualization with 30 cm spacing at robot (a) leader and (b) follower (Video demonstration available at: https://bit.ly/Swarm30cm)



(a) (b)

Figure 13. Square formation test visualization with 10 cm spacing at robot (a) leader and (b) follower (Video demonstration available at: https://bit.ly/Swarm10cm)



(a) (b)

Figure 14. Square formation test visualization with 20 cm spacing at robot (a) leader and (b) follower (Video demonstration available at: https://bit.ly/Swarm20cm)

Figures 11–14 show stable and consistent robot motion throughout the formation process. Minor errors observed under certain conditions do not lead to significant oscillations or system instability, confirming the effectiveness and reliability of the integrated PID control, RPM imbalance correction, and odometry system.

## IV. CONCLUSION

This study develops a differential-drive motor speed control system for swarm robots using DC motors and encoder sensors by integrating a Proportional–Integral–Derivative (PID) controller with an RPM imbalance correction method and odometry-based evaluation. Motor characterization shows inter-motor speed variations below 5%, allowing effective compensation through the proposed control strategy. Testing of three PID parameter sets identified an optimal configuration that ***ensured*** stable motor speed control. The RPM imbalance correction maintained motor speeds within 49–51 RPM for a 50 RPM target, achieving errors below 1% while reducing overshoot and settling time.

Formation experiments using triangular and square patterns demonstrated average position and orientation errors below 2%, confirming reliable coordination and formation stability. Overall, the combined PID and RPM imbalance correction approach improves motor speed stability and motion accuracy, showing strong potential for multi-robot applications requiring precise coordination. Future work will focus on enhancing position estimation using additional sensors such as IMUs, applying adaptive or intelligent control methods, and extending validation to larger robot groups and more complex formation and obstacle-avoidance scenarios.

## REFERENCES

[1] A. Kyzyrkanov, N. Tursynova, D. Yedilkhan, Z. Otarbay, S. Atanov, and S. Aljawarneh, “Intelligent coordination for a swarm of autonomous mobile robots,” in *Proc. 1st Int. Workshop on Intelligent Mobile Systems based on Internet of Things*, Marshall University, Huntington, WV, USA, 2024.

[2] S. Samsiana, P. W. Agung S., and R. Anwar, “Sistem odometry pada robot roda dua dengan penggerak diferensial,” in *Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung (SENTER 2020)*, Universitas Islam 45 Bekasi, Indonesia, 2020.

[3] V. Basiuk, S. Maksymova, O. Chala, and O. Miliutina, “Mobile robot position determining using odometry method,” *International Journal of Engineering Research*, vol. 3, no. 3, 2023.

[4] F. Suryatini, A. S. Sunarya, and Q. K. ’Amilin, “Optimasi kendali PID kecepatan motor DC dengan metode tuning particle swarm optimization,” *JITEL (Jurnal Ilmiah Telekomunikasi, Elektronika, dan Listrik Tenaga)*, vol. 5, no. 1, 2025.

[5] M. Ferreira, L. Moreira, and A. Lopes, “Differential drive kinematics and odometry for a mobile robot using TwinCAT,” *ERA*, vol. 31, no. 4, 2023.

[6] I. A. Hameed, L. H. Abbud, J. A. Abdulsaheb, A. T. Azar, M. Mezher, J. A. Jawad, W. R. Abdul-Adheem, I. K. Ibraheem, and N. A. Kamal, “A new nonlinear dynamic speed controller for a differential drive mobile robot,” *JITEKI*, vol. 10, no. 3, 2024.

[7] E. Rakhman, N. C. Basjaruddin, and V. E. P. Susanto, “Robot mobile otonom menggunakan metode odometry,” *Jurnal Teknik Elektro*, Politeknik Negeri Bandung, 2022.

[8] A. Khumaidi, M. A. Yaqin, R. Y. Adhitya, S. M. Irsyad, and M. J. Saputra, “Penerapan rotary encoder sebagai sistem odometry untuk navigasi robot sepak bola beroda,” *JURITEK: Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer*, vol. 5, no. 3, 2024.

[9] H. Khan, S. Khatoon, P. Gaur, M. Abbas, C. A. Saleel, and S. A. Khan, “Speed control of wheeled mobile robot by nature-inspired social spider algorithm-based PID controller,” *Journal of Engineering Science and Technology Review*, vol. 13, no. 3, 2024.

[10] O. O. Martins, A. A. Adekunle, S. B. Adejuyigbe, O. H. Adeyemi, and M. O. Arowolo, “Wheeled mobile robot path planning and path tracking controller algorithms: a review,” *Journal of Engineering Science and Technology Review*, vol. 13, no. 3, 2020.

[11] C. De Giorgi, D. De Palma, and G. Parlangeli, “Online odometry calibration for differential drive mobile robots in low traction conditions with slippage,” *Robotics*, vol. 13, no. 7, 2024.

[12] H. Supriyono, F. F. Alanro, and A. Supardi, “Development of DC motor speed control using PID based on Arduino and MATLAB for laboratory trainer,” *Jurnal Nasional Teknik Elektro*, vol. 13, no. 1, 2024.

[13] M. D. I. Putri, A. Ma’arif, and R. D. Puriyanto, “Pengendali kecepatan sudut motor DC menggunakan kontrol PID dan tuning Ziegler Nichols,” *Techno: Jurnal Teknik*, vol. 23, no. 1, 2022.

[14] G. Shadrin, A. Krasavin, G. Nazenova, A. Kussaiyn-Murat, A. Kadyroldina, T. Haidegger, and D. Alontseva, “Application of compensation algorithms to control the speed and course of a four-wheeled mobile robot,” *Sensors*, vol. 24, 2024.

[15] M. S. P. Fahmizal, M. S. Pratikno, H. N. Isnianto, A. Mayub, H. Maghfiroh, and P. Anugrah, “Control and navigation of differential drive mobile robot with PID and Hector SLAM: simulation and implementation,” *JITEKI*, vol. 10, no. 3, 2024.

**Logbook Project RE-701 Minggu ke 01**

tgl. 25-09-2025s/d tgl 30-09-2025

**Identitas Mahasiswa**

Nama Mahasiswa : Alexander Cang

NIM : 4222201027

**Identitas PBL dan Pembimbing**

Kode PBL : PBL-RE-049

Nama PBL : Swarm Robot

Pembimbing : Anugerah Wibisana, S.Tr.T., M.Tr.T.

**Rincian Kegiatan Pelaksanaan Projek**

**Bagian saya adalah bagian sistem kendali pada robot atau (sestem kontrol)**
**Untuk progress yang sudah saya buat adalah dimana saya memcoba analisa dengan target yang sudah saya dapat pada semester lalu dan dimana sama membuat pengujian sebelum/sesudah penerapan PID dan dari data yang saya coba ambil adalah dengan data yang sudah saya dapatkan saya melakukan percobaan kembali dimana saya mendapatkan saran dari manpro untuk terus melakukan pengujian dengan bergerakan robot lurus dan lihat kondisi robot setelah selesai maka lanjut dengan tahap segi 4 dan segi 3. Tujuan dari penerapan PID kedalam robot adalah untuk menstabilkan pergerakan robot agar robot bergerak dengan lurus dan tidak membelok pada saat berjalan lurus.**
**Berikut ini adalah hasil dokumentasi saya dan data yang sudah saya ambil sebelumnya :**
*(https://drive.google.com/drive/u/0/folders/1xUOtX6TtwwcrQdpBoOxOdI1HZK6VIzbV )*

Batam, 30-09-2025

Pembimbing

( Anugerah Wibisana, S.Tr.T., M.Tr.T.)

**Logbook Project RE-701 Minggu ke 01**

tgl. 02-10-2025s/d tgl 09-10-2025

**Identitas Mahasiswa**

Nama Mahasiswa : Alexander Cang

NIM : 4222201027

**Identitas PBL dan Pembimbing**

Kode PBL : PBL-RE-049

Nama PBL : Swarm Robot

Pembimbing : Anugerah Wibisana, S.Tr.T., M.Tr.T.

**Rincian Kegiatan Pelaksanaan Projek**

**Bagian saya adalah bagian sistem kendali pada robot atau system kontrol**
**Pada week kedua ini saya mencoba tunning kembali pada robot 1 dimana pada robot satu ini saat mencoba menjalankan system path planning memiliki problem saat jalan tidak mulus atau membelok kiri, kanan. Setelah saya check hasil pergerakan robot saya mendapatkan satu situasi dimana saat robot saya coba bergerak lurus robot masih bisa membelok dikarenakan adanya data yang tidak nornal pada saat target yang saya set nilai RPM dan juga kelihatan dari video yang saya ambil.**
**Berikut ini adalah hasil dokumentasi saya dan data yang sudah saya ambil sebelumnya :**
*( https://drive.google.com/drive/u/0/folders/1EaTIUu8s-IlgLkyaHbM-QzCI6TAo2SBn )*

Batam, 12-10-2025

Pembimbing

( Anugerah Wibisana, S.Tr.T., M.Tr.T.)

**Logbook Project RE-701 Minggu ke 01**

tgl. 09-10-2025s/d tgl 16-10-2025

**Identitas Mahasiswa**

Nama Mahasiswa : Alexander Cang

NIM : 4222201027

**Identitas PBL dan Pembimbing**

Kode PBL : PBL-RE-049

Nama PBL : Swarm Robot

Pembimbing : Anugerah Wibisana, S.Tr.T., M.Tr.T.

**Rincian Kegiatan Pelaksanaan Projek**

**Bagian saya adalah bagian sistem kendali pada robot atau (sestem kontrol)**
**Pada minggu ketiga ini saya mencoba untuk riset menggunakan sensor gyro tipe GY25 dimana untuk membaca posisi arah hadap pada robot dan plan saya mencoba sensor tersebut dengan tujuan saya ingin menguji pid saya dengan membuat formasi bentuk segi 4, segi 3, bintang. Alasan saya ingin membuat formasi tersebut untuk mengetahui tinggkat akurasi pada parameter pid saya dan pada probelm yang saya hadapi adalah kekurangan pin pada mikrokontroller dimana pada mikrokontrolller ini pin I2C dan pin RX, TX sudah digunakan pada sensor encoder yang ada dimotor sehingga pin yang perlu saya gunakan tersebut sudah tidak bisa digunakan lagi dan jika saya ingin menggunakan multiplexer modul, kekurangan nya pada multiplexer tersebut malah tidak ada pin interrupt sehingga saya tidak melanjutkan mengambil data pada sensor tersebut dan saya berali ke data encoder.**
**Berikut ini adalah hasil dokumentasi saya dan data yang sudah saya ambil sebelumnya :**
*(*https://drive.google.com/drive/folders/1Y2Q6S6vvzoBeSajQURZkzLjATz-drY8_?hl=id*)*

Batam, 23-10-2025

Pembimbing

( Anugerah Wibisana, S.Tr.T., M.Tr.T.)

**Logbook Project RE-701 Minggu ke 04**

tgl. 22-09-2025s/d tgl 26-09-2025

**Identitas Mahasiswa**

Nama Mahasiswa : Alexander Cang

NIM : 4222201027

**Identitas PBL dan Pembimbing**

Kode PBL : PBL-RE-049

Nama PBL : Swarm Robot

Pembimbing : Anugerah Wibisana, S.Tr.T., M.Tr.T.

**Rincian Kegiatan Pelaksanaan Projek**

**Bagian saya adalah bagian sistem kendali pada robot atau (sestem kontrol)**
**Pada minggu ke empat saya melanjutkan pengujian pad robot saya di mana saya mengambil data kestabilan robot saya dengan menguji robot berjalan lurus dengan set parameter PID dengan proses tunning manual dan kebetulan pada semester lalu saya mendapatkan kritikan dan saran dari dosen pengujia benchmark final di mana dosen pengujia menanyakan hasil data yang saya buat dan parameter yang saya dapat bagaimana cara mengetahui tingkat kestabilitasnya sedangkan saya hanya menampilkan satu data terakhir dan kama dari itu saya di sarankan mengambil data dengan step -step dari awal hingga menghasilkan data dan hasil yang bagus.**

**https://drive.google.com/drive/folders/18R20thDQ1rtYkklo7YMKFgPAkJ9fGBTz?hl=id**

Batam, 20-12-2025

Pembimbing

( Anugerah Wibisana, S.Tr.T., M.Tr.T.)

**Logbook Project RE-701 Minggu ke 05**

tgl. 29-09-2025s/d tgl 03-10-2025

**Identitas Mahasiswa**

Nama Mahasiswa : Alexander Cang

NIM : 4222201027

**Identitas PBL dan Pembimbing**

Kode PBL : PBL-RE-049

Nama PBL : Swarm Robot

Pembimbing : Anugerah Wibisana, S.Tr.T., M.Tr.T.

**Rincian Kegiatan Pelaksanaan Projek**

**Bagian saya adalah bagian sistem kendali pada robot atau (sestem kontrol)**
**Pada minggu ke lima pada minggu kelima ini saya mengambil data sample dengan masing 50 sample di mana untuk menguji kestabilan pergerakan robot dari perbandingan data robot dan sekalian saya membuat perbandingan antara data test 1 , 2, 3, dan 4. saya memdapati data yang di mana bisa membuat pengujian data kestabilan pergerakan robot saya.**

**https://docs.google.com/spreadsheets/d/13glmJxe6aAFAk3w6vaEwbiLYwMJGYqn5/edit?gid=661531445#gid=661531445**

Batam, 20-12-2025

Pembimbing

( Anugerah Wibisana, S.Tr.T., M.Tr.T.)

**Logbook Project RE-701 Minggu ke 06**

tgl. 06-10-2025s/d tgl 10-10-2025

**Identitas Mahasiswa**

Nama Mahasiswa : Alexander Cang

NIM : 4222201027

**Identitas PBL dan Pembimbing**

Kode PBL : PBL-RE-049

Nama PBL : Swarm Robot

Pembimbing : Anugerah Wibisana, S.Tr.T., M.Tr.T.

**Rincian Kegiatan Pelaksanaan Projek**

**Bagian saya adalah bagian sistem kendali pada robot atau (sestem kontrol)**
**Pada minggu ke enam saya melanjutkan pengujian odometri pada robot saya di mana pada pergerakan robot saya menambahkan pergerakan dimana saya mencoba menjalankan robot dengan taeget tertentu teru robot berputar arah 90deg setelah itu robot stop, nah di sini saya mendapati robot berputar arah dengan tidak stabil karena robot masih belum sempurna dengan taeget yang saya berikan.**

**https://drive.google.com/drive/folders/1iIknENXjOG4gCjBfsU8u9ndBfk1G9Yed?hl=id**

Batam, 20-12-2025

Pembimbing

( Anugerah Wibisana, S.Tr.T., M.Tr.T.)

**Logbook Project RE-701 Minggu ke 07**

tgl. 13-10-2025s/d tgl 17-10-2025

**Identitas Mahasiswa**

Nama Mahasiswa : Alexander Cang

NIM : 4222201027

**Identitas PBL dan Pembimbing**

Kode PBL : PBL-RE-049

Nama PBL : Swarm Robot

Pembimbing : Anugerah Wibisana, S.Tr.T., M.Tr.T.

**Rincian Kegiatan Pelaksanaan Projek**

**Bagian saya adalah bagian sistem kendali pada robot atau (sestem kontrol)**
**Pada minggu ke tujuh karena hasil pengujian odometri saya masih tidak bagus maka saya mengulang untuk kalibrasi odometri nya Kembali di mana hasil kalibrasinya masih belum pas pada parameter odometri dan saya mencoba Kembali di mana dengan mengambil data dengan mutar roda robot secara manual hingga mengeluarkan hasil yang sempurna hingga bisa digunakan untuk pengujian selanjutnya.**

**https://drive.google.com/drive/folders/1KKWE-p6_n5hAFK5lMyZVxb2MUwzpTRdv?hl=id**

Batam, 20-12-2025

Pembimbing

( Anugerah Wibisana, S.Tr.T., M.Tr.T.)

**Logbook Project RE-701 Minggu ke 08**

tgl. 20-10-2025s/d tgl 27-10-2025

**Identitas Mahasiswa**

Nama Mahasiswa : Alexander Cang

NIM : 4222201027

**Identitas PBL dan Pembimbing**

Kode PBL : PBL-RE-049

Nama PBL : Swarm Robot

Pembimbing : Anugerah Wibisana, S.Tr.T., M.Tr.T.

**Rincian Kegiatan Pelaksanaan Projek**

**Bagian saya adalah bagian sistem kendali pada robot atau (sestem kontrol)**
**Pada minggu ke delapan ini saya melanjutkan pengujian odometri dengan menguji penerapan target robot maju dan mencapai target dan juga penerapan fuzzy balance pada pid yang saya ambil yang dimana fuzzy balance berfungsi sebagai mengatur output dari pid yang di mana masih belum cukup baik dengan hanya mengandalkan pid saja berikut adalah hasilnya :**

**https://drive.google.com/drive/u/0/folders/1Nq1oAndH0TNtHOq4eGw-90fRw_899TTg**

Batam, 10-01-2025

Pembimbing

( Anugerah Wibisana, S.Tr.T., M.Tr.T.)

**Logbook Project RE-701 Minggu ke 09**

tgl. 28-10-2025s/d tgl 03-11-2025

**Identitas Mahasiswa**

Nama Mahasiswa : Alexander Cang

NIM : 4222201027

**Identitas PBL dan Pembimbing**

Kode PBL : PBL-RE-049

Nama PBL : Swarm Robot

Pembimbing : Anugerah Wibisana, S.Tr.T., M.Tr.T.

**Rincian Kegiatan Pelaksanaan Projek**

**Bagian saya adalah bagian sistem kendali pada robot atau (sestem kontrol)**
**Pada minggu ke Sembilan ini dimana setelah saya ambil data target dari odometri saya lanjut mengambil data drajat (theta) dimana data tersebut akan digunakan sebagai mengetahui tingkat akurasi atau karakteristik pada setiap robot dan dengan mengambil data theta bisa mnegetahui penerapan untuk bentuk formasi pada robot dan berikut hasil pengujian nya :**

**https://drive.google.com/drive/u/0/folders/1_xhLznQFSWjf1cZabrWHMRmErUaPdEB7**

Batam, 20-12-2025

Pembimbing

( Anugerah Wibisana, S.Tr.T., M.Tr.T.)

**Logbook Project RE-701 Minggu ke 10**

tgl. 04-11-2025s/d tgl 17-11-2025

**Identitas Mahasiswa**

Nama Mahasiswa : Alexander Cang

NIM : 4222201027

**Identitas PBL dan Pembimbing**

Kode PBL : PBL-RE-049

Nama PBL : Swarm Robot

Pembimbing : Anugerah Wibisana, S.Tr.T., M.Tr.T.

**Rincian Kegiatan Pelaksanaan Projek**

**Bagian saya adalah bagian sistem kendali pada robot atau (sestem kontrol)**
**Pada minggu ke kesepuluh ini dimana saya mencoba penerapan formasi dengan bentuk segi 4 dan segi tiga dan dengan mengetahui akurasi dari implementasi pada robot yang didapatkan dari gabungan pid fuzzy dan odometri dan dengan pengujian ini saya bisa mendapatkan data hasil perbandingan robot leader dan robot follower berikut adalah hasil penerapannya :**

**https://drive.google.com/drive/u/0/folders/1q8-2Nx9ygbcclwHWHEI4Hq02TWBwMIab**

Batam, 20-12-2025

Pembimbing

( Anugerah Wibisana, S.Tr.T., M.Tr.T.)