Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 5, Special Issue "COVID-19: Emerging Research", 2021

PID-based with Odometry for Trajectory Tracking Control on Four-wheel Omnidirectional Covid-19 Aromatherapy Robot

Iswanto [1, 2*], Alfian Ma'arif [3], Nia Maharani Raharja [4], Gatot Supangkat [2, 5], Fitri Arofiati [6], Ravi Sekhar [7], Dhiya Uddin Rijalusalam [1]

[1] Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta, Yogyakarta 55183, Indonesia

[2] Department of Engineer Profession Program, Universitas Muhammadiyah Yogyakarta, Yogyakarta 55183, Indonesia

[3] Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia

[4] Department of Electrical Engineering, Universitas Negeri Sunan Kalijaga, Yogyakarta 55281, Indonesia

[5] Department of Agrotechnology, Universitas Muhammadiyah Yogyakarta, Yogyakarta 55183, Indonesia

[6] Department of Nursing Master, Universitas Muhammadiyah Yogyakarta, Yogyakarta 55183, Indonesia

[7] Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India

Abstract

Inhalation therapy is one of the most popular treatments for many pulmonary conditions. The proposed Covid-19 aromatherapy robot is a type of Unmanned Ground Vehicle (UGV) mobile robot that delivers therapeutic vaporized essential oils or drugs needed to prevent or treat Covid-19 infections. It uses four omnidirectional wheels with a controlled speed to possibly move in all directions according to its trajectory. All motors for straight, left, or right directions need to be controlled, or the robot will be off-target. The paper presents omnidirectional four-wheeled robot trajectory tracking control based on PID and odometry. The odometry was used to obtain the robot's position and orientation, creating the global map. PID-based controls are used for three purposes: motor speed control, heading control, and position control. The omnidirectional robot had successfully controlled the movement of its four wheels at low speed on the trajectory tracking with a performance criterion value of 0.1 for the IAEH, 4.0 for MAEH, 0.01 for RMSEH, 0.00 for RMSEXY, and 0.06 for REBS. According to the experiment results, the robot's linear velocity error rate is 2%, with an average test value of 1.3 percent. The robot heading effective error value on all trajectories is 0.6%. The robot's direction can be monitored and be maintained at the planned trajectory.

Keywords:

PID Control;
Odometry;
Four Wheel Omni Directional;
Aromatherapy Robot;
Trajectory Tracking Control;
Feedback Control.

Article History:

Received:	04	July	2021
Revised:	09	October	2021
Accepted:	21	October	2021
Published:	15	November	2021

1- Introduction

The increasing number of Covid-19 patients has made researchers develop technology in the medical field to treat Covid-19 patients. Many previous researchers have done research on therapy for patients. For example, antiplatelet therapy in Covid-19 patients was studied by Banik [1]. Antithrombotic therapy in patients with Covid-19 was studied by Godino [2]. The clinical impact of pre-admission antithrombotic therapy on hospitalized patients with Covid-19 was studied by Russo [3]. Low-dose radiation therapy has been researched to treat post-Covid immune-associated pneumonia

* **CONTACT**: iswanto_te@umy.ac.id

DOI: http://dx.doi.org/10.28991/esj-2021-SPER-13

© 2021 by the authors. Licensee ESJ, Italy. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC-BY) license (https://creativecommons.org/licenses/by/4.0/).

[4]. Nutritional therapy for Covid-19 patients was found to reduce malnutrition risk and improves patient clinical outcomes [5]. Plasma therapy for seriously ill and critically ill patients diagnosed with Covid-19 was also researched [6]. Ghasemiyah researched pharmacokinetic and pharmacodynamic aspects of drug therapy in patients with moderate to severe infection [7]. Tocilizumab therapy for COVID-19 was also studied by Kaminski [8].

Among many therapy treatments, one of the most popular is inhalation therapy. Gowrishankar conducted an in silico study on potential phytochemicals in traditional Indian herbal steam inhalation therapy to combat Covid-19 [9]. On the other hand, the safety and efficacy of ozone therapy in mild to moderate Covid-19 patients were studied by Shah [10]. It is also known that inhalation is the most preferred route of administration for the treatment of many pulmonary conditions [11]. Hence, aromatherapy robot is a promising technology to be developed, which can be utilized to carry out inhalation delivery. A Covid aromatherapy robot is a mobile robot that is proposed to bring therapeutic essential oils or drugs needed to carry out therapy to prevent or treat Covid-19 infections.

A mobile robot can typically move, not limited to a single location [12]. The mobile robot's distinguishing feature is its actuator in the form of a wheel that pushes the entire robot body, allowing it to shift positions. However, one of the issues with mobile robots is the robot's mobility. Most wheeled mobile robots have ordinary wheels, making their travel very limited [13]. Ackerman steering and differential steering are two types of robot navigation that frequently encounter low mobility problems. Both can move forward and veer but cannot move in all directions, putting them in the category of non-holonomic robots. The limited directional movement will obstruct the robot's ability to travel in different directions. This is not ideal to be used on the Covid aromatherapy robot, since it is hoped to be used in many different environments such as hospitals, houses of self-isolated patients, emergency shelters, or just public places in general.

Hence, an Omni-wheeled robot is used since it can travel more freely and safely to avoid hitting obstacles or walls [14], which are found in hospitals or public places. Ideally, the robot should be able to reach all directions without having to switch or change directions [15]. The robot should be able to move forward and backward; it should also move in all directions in the Cartesian x-y plane without rotating its body. The Omnidirectional robot is suitable for this function since it is a holonomic robot that can move in any direction without having to change its orientation [16].

The Omnidirectional robot has several advantages over other wheeled robots, including the ability to move in all directions with minimum turning radius and great traction movement than other drivers such as Mecanums or other differential wheels. This wheel is designed to look like a regular wheel but has sub-wheels that are 90 degrees apart from the outer wheels. As a result, the mechanical device helps it to maneuver without shifting its body position.

The wheel has a complex mathematical model to drive the robot to the desired location. In general, this mathematical model involves robot modeling control, specifically inverse and advanced kinematics. The configuration of three wheels and four wheels are common designs of Omni-wheeled robots [17]. Four-wheeled robots are more common due to their easiness to be maneuvered than three-wheeled robots.

Besides the configuration of the wheels, external and internal sensors also need to be appropriately configurated and chosen. External and internal sensors are used to drive the robot. External sensors, such as line and wall sensors, are used to direct the robot's movement. The robot can travel along lines and walls using these sensors. Internal sensors, such as a location sensor in the form of rotary encoder sensors, are used to move the robot towards the target [18]. Data obtained from these sensors are used in the odometrical measurement to assess the robot's position and orientation [19].

Three variables that were derived from odometrical measurements affect the robot's location and orientation on the coordinate axis framework. The first variable is the X and Y position, which signifies the rear axle's center point. The second variable is the angle between the mobile robot's body to the X-axis. Meanwhile, the third variable is the angle of the mobile robot's steering wheel.

The odometrical control method has been widely used to solve the robot trajectory tracking problem [20]. Trajectory tracking control is an autonomous mobile robot control method that allows the robot to follow a predetermined pattern. The methods of controlling trajectory tracking are categorized into two: offline and online. The offline trajectory tracking control determines the robot's location, speed [21], and orientation using an odometrical sensor. Meanwhile, the online trajectory tracking control determines the location and orientation of the robot using proximity sensors, including camera sensors, lidar sensors, and ultrasonic sensors. Odometrical control has more advantages than kinematic control and other control algorithms, especially for trajectory tracking. Besides PID control, odometrical control has the simplest mathematical calculation, which leads to a fast computation process.

Some researchers have applied trajectory tracking control in various applications. For example, trajectory tracking control has been applied to multi-robot [22, 23]. Many applications also have been found in UAV [24-27]. Still, in robotic fields, some researchers also have applied trajectory tracking control for Robot Manipulator [28], Magnetic Microrobot [29], Soft Robots Interacting [30], and Laparoscopic 4-DoF robot [31]. Similar to the proposed robot design, some researchers also have been applied the trajectory tracking control for wheeled robots [32]–[34]. Moreover, some complex control algorithms also had been used for trajectory tracking control, such as predictive control model [13], Fuzzy [26, 35], Curve Algorithm [36]. Even some researchers have applied the advanced Sliding Mode Controller [37-39]. PID control also has been applied for trajectory tracking control [40], and its combination with another control algorithm such as in Neural Adaptive PID [41].

This paper presents the Covid Aromatherapy robot, which is an omnidirectional wheel robot with a four-wheel configuration with rotary encoders and diffusers. Regarding its specific function, this research specifically proposes trajectory tracking control on the robot based on PID with odometry and evaluates its performance based on the performance evaluation criterion. PID control, which is used in omnidirectional robots for trajectory tracking control, is explored to support the control system design. In the trajectory tracking control, PID-based controls are used for three purposes: motor speed control, heading control, and position control.

The structure of the paper is designed as follows. The first section is the introduction. Next, the second section is the methodology, which consists of several sub-sections such as kinematic modeling, inverse kinematics, forward kinematics, odometry system, and PID control technique. The next section explains the control system architecture, which consists of explanations about the robot's trajectory planning, PID implementation for trajectory planning, PID implementation for robot movement, PID implementation for robot heading, the overall system architecture design, and the parameters of performance evaluation. The fourth section is the experiment and results, which consists of detailed analysis and discussions of the experiment. Then, the fifth section is the conclusions and future works. The sixth section is the acknowledgments. The last section is the references, which lists all references used in the research.

2- Research Methodology

2-1- Kinematic Modeling

Figure 1 illustrates the display of the Omnidirectional robot's wheel positioning frame. The angle formed between the wheels and the robot reference point is denoted by $\alpha_1, \alpha_2, \alpha_3, and\ \alpha_4$ in this diagram. The angular velocity of each wheel is denoted by $\omega_w = [\omega_1\ \omega_2\ \omega_3\ \omega_4]^T$, and the linear velocity of each wheel is by $v_w = [v_1\ v_2\ v_3\ v_4]^T$.

The wheel speed value will be positive when the wheel rotates clockwise (CW) and will be negative when counterclockwise (CCW). The global reference coordinates of the robot are denoted by the letters XOY. The linear velocity and angular velocity of the robot to the XOY global coordinates are notified using $v_{(g)} = [vg_{(x)}\quad vg_{(y)}\ \ \omega\ ]$. R denotes the distance between the wheel and the robot's center, and r denotes the radius of the Omni wheel.

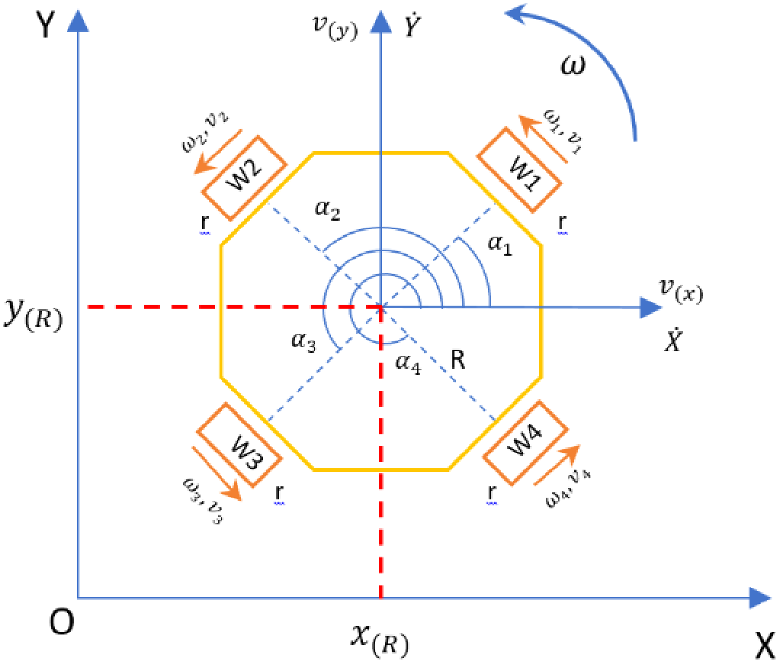


Figure 1. Wheel placement of the omnidirectional robot.

Figure 2 shows the omnidirectional robot's coordinate and orientation to the global XOY coordinates. The linear velocity of the robot relative to the body frame of the robot is denoted using $v_{(R)} = [v_{(x)}\quad v_{(y)}\ ]^T$. The robot's coordinates and orientation to global coordinates is notated by $X_{(g)} = [x_{(R)}\ y_{(R)}\ \psi]^T$. ω is the notation of the angular velocity of the robot to the global reference coordinates, $\varphi_{(R)}$ is the direction of the robot movement at global coordinates, and L is the notation of the resultant coordinates the robot has passed.

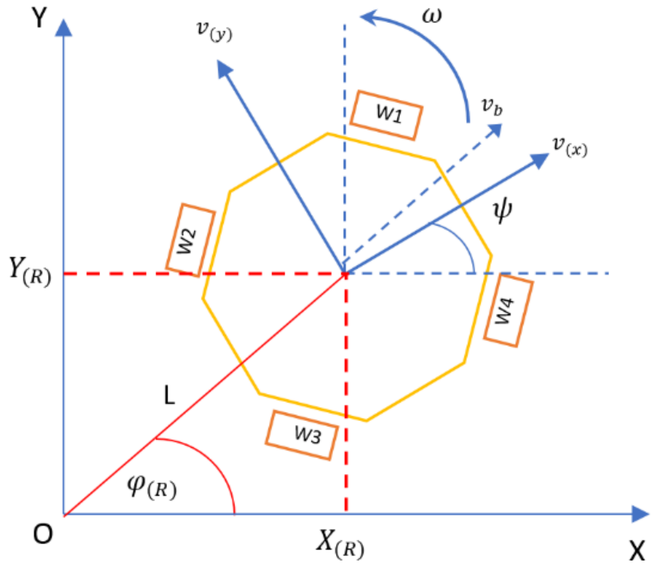


Figure 2. The coordinates and orientation of the robot to the global coordinates of XOY.

By referring to Figure 2, several equations are formulated.

$$\varphi_{(R)} = tan^{-1}\frac{Y_{(R)}}{X_{(R)}} \tag{1}$$

$$\mathrm{L} = \sqrt{{X_{(R)}}^2 + {Y_{(R)}}^2} \tag{2}$$

$$\begin{matrix} v_{(x)} \\ v_{(y)} \\ \omega \end{matrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \omega \end{bmatrix} \tag{3}$$

$$v_b = \sqrt{{v_{(x)}}^2 + {v_{(y)}}^2} \tag{4}$$

where ψ is the notation of the direction of the robot's orientation to global coordinates, $vb_{(R)}$ is the resultant linear velocity of the robot $v_{(x)}$ and $v_{(y)}$.

There are two types of kinematic models in the robot modeling system: inverse kinematics (IK) and forward kinematics (FK). The robot platform uses IK to determine the wheel's linear velocity $v_{(w)} = [v_1\ v_2\ v_3\ v_4]^T$ and FK to determine the robot's linear velocity in global coordinates $v_{(g)} = [vg_{(x)} \quad vg_{(y)}\ \omega\ ]^T$.

2-2- Inverse Kinematics Omnidirectional Robot

The linear velocity value of each wheel $v_{(w)}$ can be determined based on the changes in the linear movement of the robot to global coordinates denoted by the symbol $v_{(g)}$. By referring to Figures 1 and 2, the inverse kinematic equation is obtained as follows:

$$v_w = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = T(\theta)v_{(g)} \tag{5}$$

where $T(\theta)$ is the rotation matrix equation for each wheel in Figures 1 and 2; it can be written as Equation 6.

$$T(\theta) = \begin{bmatrix} -\sin(\psi\ + \alpha_1\ ) & \cos\ (\psi + \alpha_1\ ) & R \\ -\sin(\psi + \alpha_2\ ) & \cos\ (\psi + \alpha_2\ ) & R \\ -\sin(\psi + \alpha_3\ ) & \cos\ (\psi + \alpha_3\ ) & R \\ -\sin(\psi + \alpha_4\ ) & \cos\ (\psi + \alpha_4\ ) & R \end{bmatrix} \tag{6}$$

$$v_{(g)} = [vg_{(x)} \quad vg_{(y)} \quad \omega]^T \tag{7}$$

2-3- Forward Kinematics Omnidirectional Robot

The linear movement of the robot $v_{(g)}$ is determined based on the linear velocity value of each wheel v_w. Referring to the inverse kinematic Equation 3, the value of $v_{(g)}$ is obtained by $v_{(g)} = v_w.T(\theta)^{-1}$. However, because $T(\theta)$ in Equation 6 is a 4×3 matrix, so $T(\theta)^{-1}$ is a pseudo-inverse. A calculation error will occur if this formula is used. Hence, referring to Figures 1 and 2, a new equation is proposed as seen in Equation 8:

$$V_{(g)} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \omega \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} D(\theta) \tag{8}$$

$D(\theta)$ is the transpose of rotation matrix in Equation 6, which can be written as;

$$D(\theta) = \frac{1}{2} \begin{bmatrix} -\sin(\psi\ + \alpha_1\ ) & \cos\ (\psi + \alpha_1\ ) & 2R^{-1} \\ -\sin(\psi + \alpha_2\ ) & \cos\ (\psi + \alpha_2\ ) & 2R^{-1} \\ -\sin(\psi + \alpha_3\ ) & \cos\ (\psi + \alpha_3\ ) & 2R^{-1} \\ -\sin(\psi + \alpha_4\ ) & \cos\ (\psi + \alpha_4\ ) & 2R^{-1} \end{bmatrix}^T \tag{9}$$

where R is the distance of the wheel to the center point of the robot.

2-4- Odometry System

Odometry is a method used to obtain data from various motion sensors to estimate position changes over a specific time interval. In this analysis, an odometrical sensor was applied using an optical rotary encoder that counts the pulses of motor rotation in time units. Further, these pulses can be translated into variables, such as wheel speed, robot orientation, robot movement, and other navigation variables. It is necessary to read the linear velocity of the wheel (v_n) to obtain odometry data for changes of the robot position using Equation 10:

$$v_n = \frac{\dot{p}_{(n)}}{PPR}\ x\ K, \qquad n = 1,2,3,4 \tag{10}$$

where $v_{(n)}$ is the wheel's linear velocity, and $\dot{p}_{(n)}$ is the number of changes in the encoder pulse with time calculated using Equation 11, PPR stands for the number of pulses per revolution as determined by the optical rotary encoder specifications, K for the circumference of the wheel, and *n* for the notation of the wheel number.

$$\dot{p}_{(n)} = \frac{dp_{(n)}}{dt}, \qquad n = 1,2,3,4 \tag{11}$$

The linear velocity of the wheel is used to calculate the linear velocity value of the robot ($v_{(g)}$) using Equation 5. The results of FK equation, as in Equation 8, are used to calculate the current coordinates and orientation of the robot $(X_{(R)}, Y_{(R)}, \psi)$ to global coordinates, using new equations as follows;

$$X_{(R)}\ = \sum_{k=1}^{n} (\dot{X}_{(n)})\ \ = \dot{X}_{(k)} + \dot{X}_{(k+1)} + \ldots + \dot{X}_{(n)} \tag{12}$$

$$Y_{(R)}\ = \sum_{k=1}^{n} (\dot{Y}_{(n)})\ \ = \dot{Y}_{(k)} + \dot{Y}_{(k+1)} + \ldots + \dot{Y}_{(n)} \tag{13}$$

$$\psi\ = \sum_{k=1}^{n} (\omega_{(n)})\ \ = \omega_{(k)} + \omega_{(k+1)} + \ldots + \omega_{(n)} \tag{14}$$

2-5- PID Control

In this study, PID controls are used for three purposes. The first purpose is for controlling the motor speed value, the second is for changing the robot's destination coordinates, and the third is for controlling the robot's targeted position. Hence, the speed of wheel and the coordinates of robot's destination can be corrected and controlled according to the trajectory tracking.

PID control uses continuous feedback to determine the precision of a system [4]. The feedback is then compared with the set-point to obtain the error. Three main parameters in the controller are Proportional, Integral, and Derivative [42]. Equation 15 is the general control equation used in PID Control [43].

$$u(t) = K_p e(t) + \ K_i \int e(t)dt + K_d \frac{de(t)}{dt} \tag{15}$$

$$e(t) = \ sp - pv \tag{16}$$

where $u_{(t)}$ is the notation of the PID control output, K_p is the notation of the proportional constant, K_i is the integral constant notation, pv, K_d is the derivative constant, $e(t)$ is the difference between the set-point value, sp, and the current sensor reading value. Each combination of constant values K_p, K_i, K_d will give a different output response. Table 1 presents the response of each combination of constant values K_p, K_i, and K_d.

Table 1. Response of each combination of constant values p, i, and d.

Closed-loop Response	Rise Time	Overshoot	Setting Time	Steady State Error	Stability
Increasing K_p	Decreasing	Increasing	Small Increase	Decreasing	Degrading
Increasing K_i	Small Decrease	Increasing	Increasing	Large Decrease	Degrading
Increasing K_d	Small Decrease	Decreasing	Decreasing	Minor Change	Improved

3- Control System Architecture

3-1- Robot Control System and Trajectory Planning

The overall block diagram of the control system for the omnidirectional robot is represented in Figure 8. The system is initialized with trajectory planning, including a series of coordinates as the trajectory planning value, the target heading as the set-point heading value, and the robot's resultant velocity value (vb). Some of the data will then be converted into the robot's direction and angular velocity. The referenced speed value will be used to calculate the set-point of speed operated by the PID control using Equation 22. Meanwhile, odometrical devices can detect the change in the robot movement value over time once it starts moving. The robot's coordinates and orientation are calculated using the forward kinematic equation, as in Equation 8. The coordinate and orientation data will be used to correct the robot's direction, speed, and orientation.

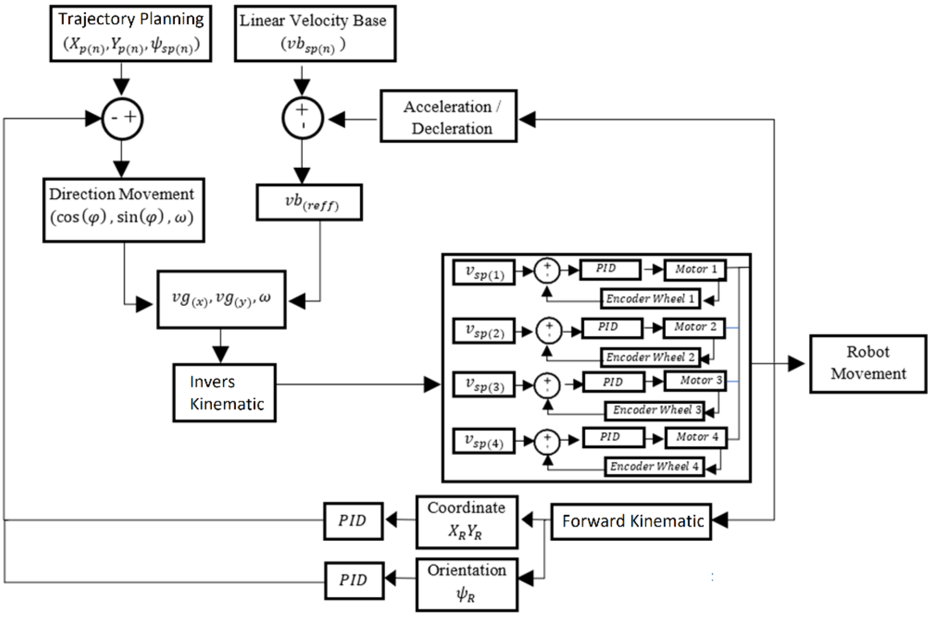
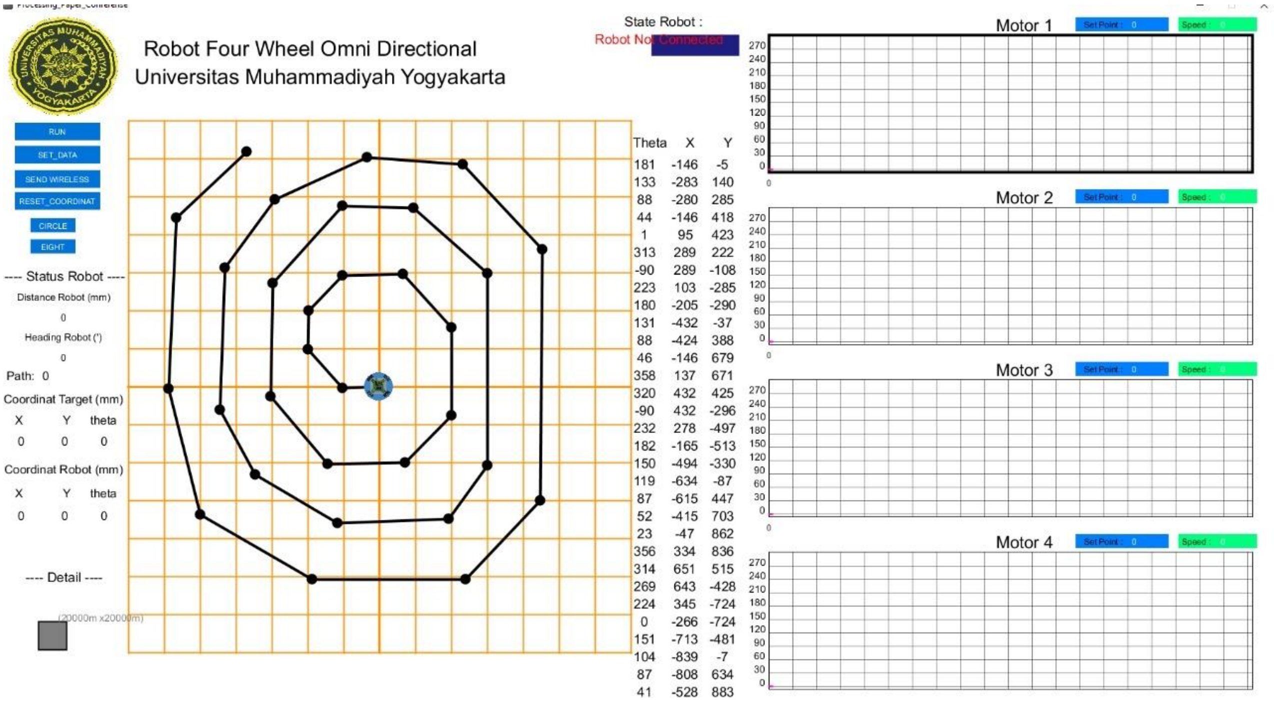


Figure 8. Control Block Diagram.

The robot movement will then be sent wirelessly to a computer by using the NRF module. A Graphical User Interface (GUI) is created to simplify the trajectory planning and movement tracking of the omnidirectional robot, as shown in Figure 4. All status of the robot movement will be displayed in the GUI.



Robot Four Wheel Omni Directional
Universitas Muhammadiyah Yogyakarta

State Robot :
Robot Not Connected

---- Status Robot ----
Distance Robot (mm)
0
Heading Robot (')
0
Path: 0

Coordinat Target (mm)

X	Y	theta
0	0	0

Coordinat Robot (mm)

X	Y	theta
0	0	0

---- Detail ----

Theta	X	Y
181	-146	-5
133	-283	140
88	-280	285
44	-146	418
1	95	423
313	289	222
-90	289	-108
223	103	-285
180	-205	-290
131	-432	-37
88	-424	388
46	-146	679
358	137	671
320	432	425
-90	432	-296
232	278	-497
182	-165	-513
150	-494	-330
119	-634	-87
87	-615	447
52	-415	703
23	-47	862
356	334	836
314	651	515
269	643	-428
224	345	-724
0	-266	-724
151	-713	-481
104	-839	-7
87	-808	634
41	-528	883

Figure 4. Graphical User Interface of the Omnidirectional Robot.

3-2- PID Implementation for Trajectory Planning

Figure 3 illustrates the basis of the robot movement control algorithm. The figure shows the robot position mapped onto the trajectory planning to be completed.

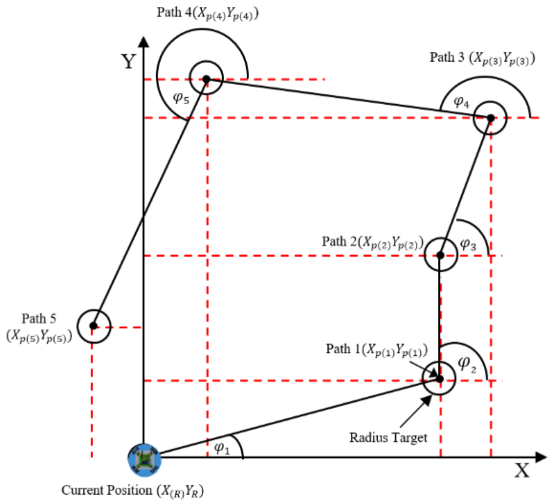

Path 4$(X_{p(4)}Y_{p(4)})$
Y
Path 3 $(X_{p(3)}Y_{p(3)})$
φ_5
φ_4
Path 2$(X_{p(2)}Y_{p(2)})$
φ_3
Path 5
$(X_{p(5)}Y_{p(5)})$
Path 1$(X_{p(1)}Y_{p(1)})$
φ_2
Radius Target
φ_1
X
Current Position $(X_{(R)}Y_R)$

Figure 3. Trajectory Planning Algorithm.

By referring to Figure 3, several equations are formulated.

$$X_t = X_{p(n)} - X_{(R)}, \tag{17}$$

$$Y_t = Y_{p(n)} - Y_{(R)}, \tag{18}$$

$$\varphi_{(n)} = tan^{-1}\frac{Y_{t(n)}}{X_{t(n)}} \tag{19}$$

$$Res = \sqrt{(X_t)^2 + (Y_t)^2} \tag{20}$$

Where $X_{t(n)}, Y_{t(n)}$ is the distance of the robot to the target position; $X_{p(n)}, Y_{p(n)}$ is the path point to the global coordinates where n denotes the sequence notation of the path point; $X_{(R)}, Y_{(R)}$ shows the current position of the ROB coordinates; φ is the direction of the robot movement, and Res is the notation of the resultant distance of the robot to the path point. The radius target shown in Figure 3 is the tolerance value of the robot where the robot is considered to have reached the specified path point value. Due to the limited accuracy of the odometry sensor used, it will be very difficult for the robot to actually reach the predefined path point. The rate of change in path points is determined by comparing the initial path point position with the target and can be calculated using the equation below.

$$e(t) = sp - pv \tag{21}$$

By identifying the direction of the robot φ as described in Equation 19, the linear velocity of the robot (vb_{reff}) can be calculated using the following equation:

$$vb_{reff} = \cos(\varphi)\, vg_{(x)} + \sin(\varphi)\, vg_{(y)} \tag{22}$$

$$vg_{(x)} = \cos(\varphi)\, Vb_{reff} \tag{23}$$

$$vg_{(y)} = \sin(\varphi)\, Vb_{reff} \tag{24}$$

where $vg_{(x)}$ and $vg_{(y)}$ are the linear velocity of the robot to the global coordinates, and vb_{reff} is the linear velocity of the robot. By entering the results of Equations 23 and 24 into the inverse kinematic Equation 5, the set-point of the linear velocity for each wheel can be calculated. The speed of each wheel was calculated using PID control, as in Equations 15 and 16.

PID control has some benefits, such as high sensitivity to respond to errors in an existing plant. However, in the case of motor speed control, this could be a disadvantage. When there is a significant change of speed, the fast control response will cause the system to be unstable with an overshoot. The steady-state also will result in an odometrical reading error due to wheel slip. For this reason, acceleration and deceleration methods are applied to the linear velocity of the robot (vb_{reff}), causing the set-point of each wheel to increase regularly and allowing the motor speed control to be monitored consistently. As shown in Equation 25, there is an alteration in the acceleration of the vb_{reff}.

$$vb_{(reff)} = \begin{Bmatrix} vb_{(reff)} = \int_{0}^{b} a(t)dt\,, & if\ vb_{reff} < \ vb_{(sp)} \\ vb_{(reff)} = vb_{(sp)}, & if\ v_{reff} \geq vb_{(sp)} \end{Bmatrix} \tag{25}$$

where b is the acceleration time duration, $a(t)$ is the time-function of acceleration. The smaller the value of a, the motor speed change is smoother, but it also takes longer to change. The set-point value of the resultant linear velocity of the robot is represented by $vb_{(sp)}$. This value is the input variable in every experiment.

When the robot reaches the target point, the motor speed will decelerate, as shown in Equation 26,

$$v_{b(reff)} = \int_{0}^{c} dc(t)\ dt\,, if\ Res < k \tag{26}$$

where c is the deceleration time, and $dc(t)$ is the time-function of deceleration. The $dc(t)$ value affects the deceleration time: the greater the value of $dc(t)$, the faster the deceleration. In Figure 3 and Equation 20, Res is the resultant distance of the robot to the path point. The value of k represents the distance notation, the point at which the robot starts to decelerate.

3-3- PID Implementation for Robot Movement

PID control is implemented to control robot movement to stay on the predetermined trajectory planning. Figures 5(a) and 5(b) shows illustrations of error in the omnidirectional robot's movement against the trajectory.

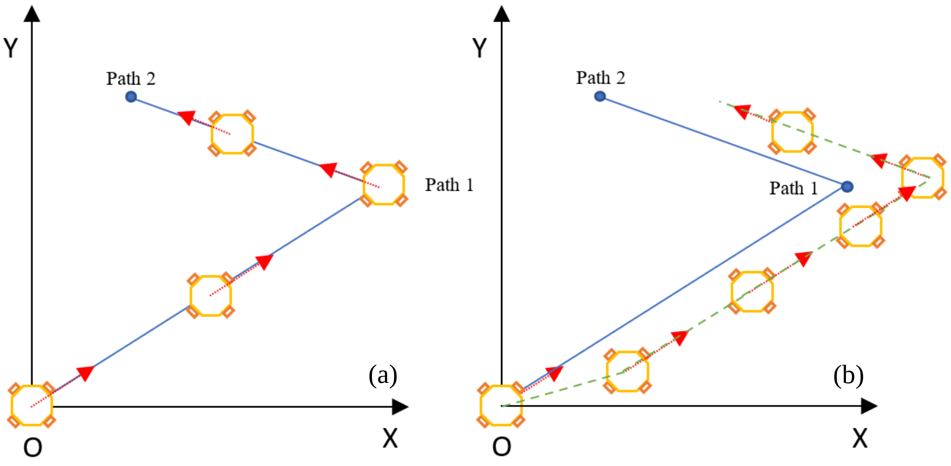


Figure 5. (a) Illustration of Ideal Robot Movement (b) Error in Robot Movement.

It is necessary to detect an error in the direction of the current robot movement φ_R in the specified trajectory planning $\varphi_{(n)}$, as displayed in Figure 6. Figure 6 illustrates a detection error of the robot's movement direction.

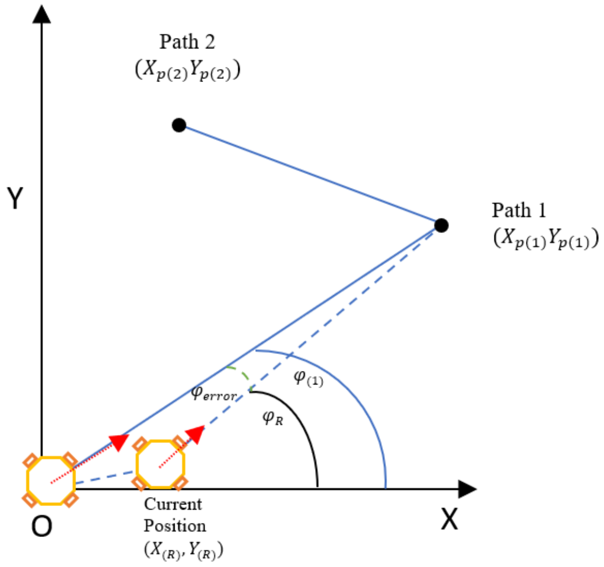


Figure 6. Error in Robot Movement Direction Detection.

By referring to Figure 6, several equations below are obtained:

$$\varphi_R = tan^{-1} \frac{Y_{p(n)} - Y_R}{X_{p(n)} - X_R} \tag{27}$$

$$\varphi_{error} = \varphi_{(n)} - \varphi_R \tag{28}$$

The value of φ_{error} is then entered into the PID equations, which are Equations 15 and 16, generating the PID controller's output. This value is then used to improve the value of the robot's direction of movement $\varphi_{(n)}$ which results in the robot moving closer to the trajectory planning value, as shown in the below equation.

$$\varphi_{correction} = \varphi_{(n)} + out\_pid_{(m)} \tag{29}$$

$out\_pid_{(m)}$ is the output of the PID for robot movement.

3-4- PID Implementation for Robot Heading

In addition to the speed and direction of robot movement, heading control is an important aspect to consider in omnidirectional robot motion control. This is because changing the robot heading will cause it to travel in the wrong direction. Figure 7 provides an example of a movement error robot triggered by a change in the robot heading direction.

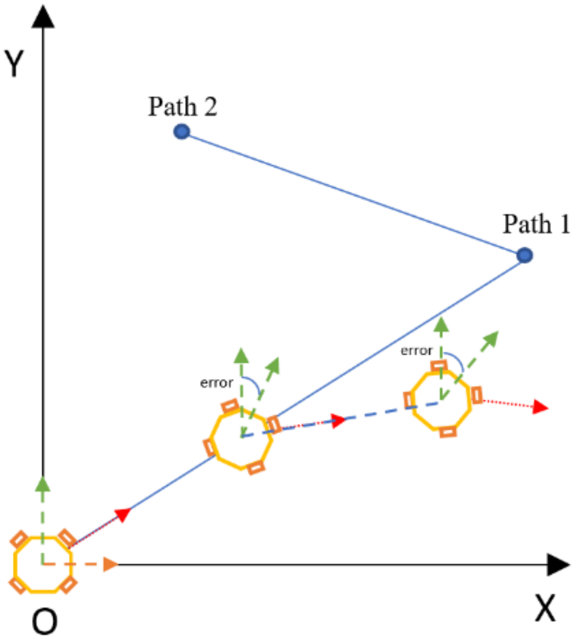


Figure 7. Illustration of movement error caused by the change of robot heading.

The error value of the robot heading is calculated using odometrical data by comparing the robot heading value ψ obtained from the FK Equation 8 to the heading set-point value $\psi_{(sp)}$, as shown in Equation 30.

$$\psi_{error} = \psi_{(sp)} - \psi_R \tag{30}$$

This value is then entered into the Equation 15 so that the PID output can correct the direction of the robot heading (ψ_R).

3-5- Overall System Architecture Design

Figure 9 illustrates a schematic diagram for the omnidirectional robot. The robot uses an ARM Cortex-M7 microcontroller for control purposes. The microcontroller then reads the encoder sensor. The microcontroller's output is fed into the motor driver to recognize the speed of the DC motor. This omnidirectional robot is equipped with communication support using the NRF module. The data from the robot is transmitted to the computer for tracking purposes.

Figure 10 shows the four-wheel omnidirectional robot. The Covid Aromatherapy Robot consists of four sets of a mini diffuser; each consists of a liquid container, four micro nozzles, a DC cooling fan, and a fogmaker. The diffuser system is automatically on when the omnidirectional robot system is on. The liquid container will carry out distilled water that has been added with therapeutic essential oils, and the fogmaker will atomize the mixture. The vapor will be sprayed outside through the nozzles to deliver the inhalation therapy. The cooling fan will control the temperature inside the container and spread out the vaporized water outside.

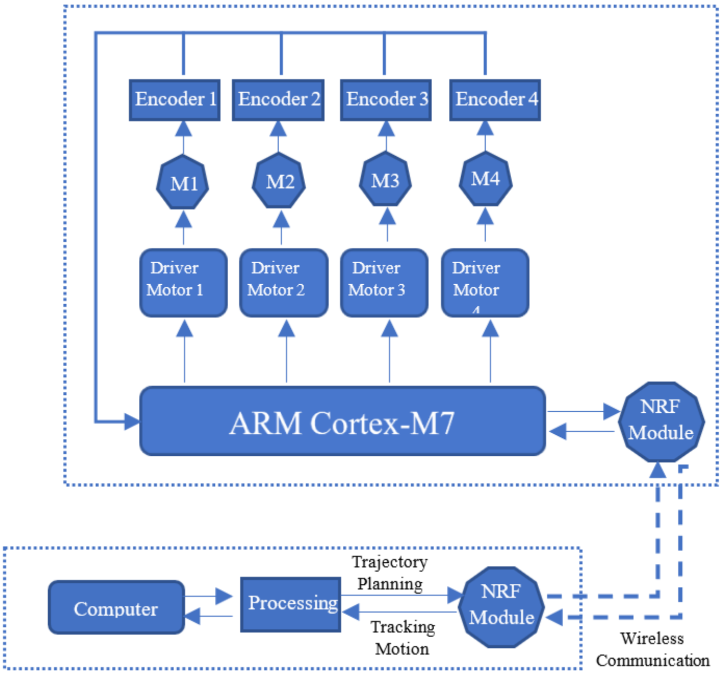


Figure 9. Electronic Wiring Diagram.

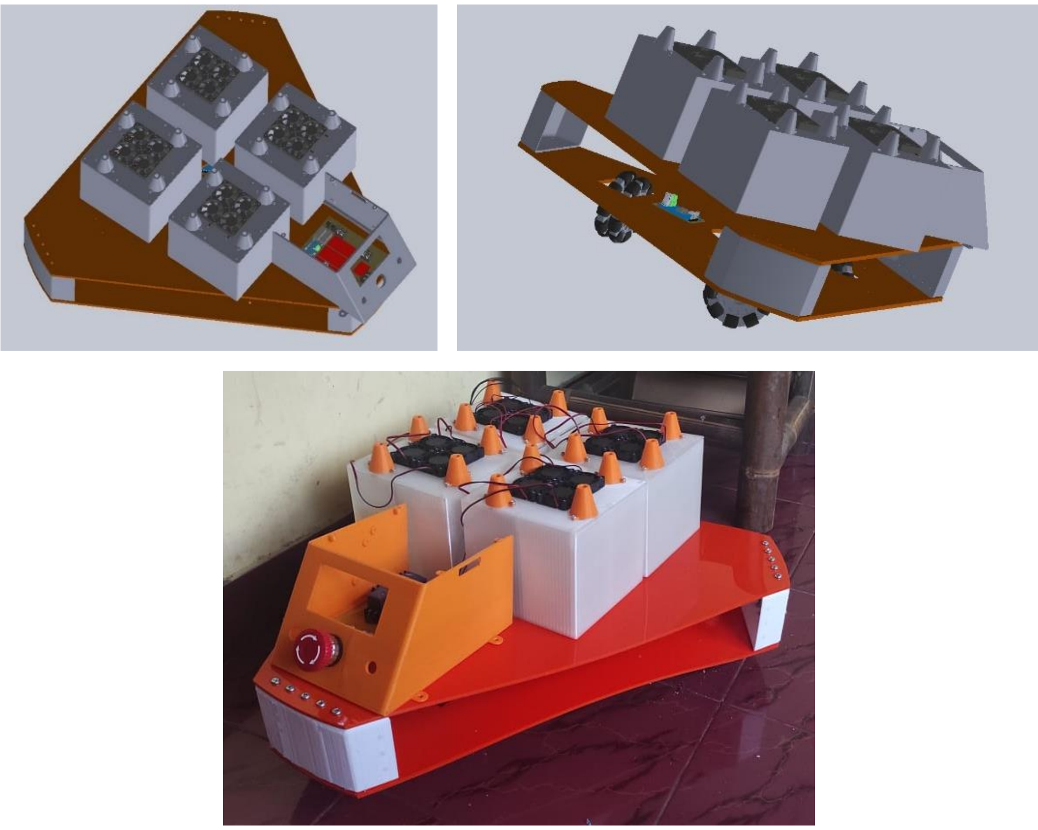


Figure 10. Four-wheel Omnidirectional Robot.

3-6- Parameters of Performance Evaluation

The following data collection approaches were used in the study to evaluate the performance of the robot movement tracking:

1- Integral of Absolute Errors. IAE_{xy} and IAE_D show the accumulated error of several set-point inputs during the robot movement time.

$$IAE_{xy} = \int_0^T (|X_e(t)| + |Y_e(t)|)\, dt \quad (31)$$

$$IAE_D \quad = \int_0^T \psi_e\,(t)\,dt \tag{32}$$

2- Maximum Absolute Errors. MAE_{xy} and MAE_H show the maximum error value from several set-points during the robot movement time.

$$MAE_{x,y} = \max X_e(t), \max Y_e(t) \tag{33}$$

$$t\,\epsilon\,[0,T]\; t\,\epsilon\,[0,T] \tag{34}$$

$$MAE_H = \max \psi_e\,(t) \tag{35}$$

$$t\,\epsilon\,[0,T] \tag{36}$$

3- Root Mean Square Errors. $RMSE_{xy}$ and $RMSE_H$ show the effective value of the error of several set-points during the robot movement time,

$$RMSE_{xy} = \sqrt{\frac{1}{H}\int_0^T (|X_e(t)| + \,|Y_e(t)|)^2\,dt} \tag{37}$$

$$RMSE_H = \sqrt{\frac{1}{H}\int_0^T [\psi_e(t)]^2\,dt} \tag{38}$$

in which H is the notation of the amount of data sampling taken

4- Rate Error of Base Speed (RE_{BS}). It shows the error rate (%) of the speed control of the four wheels while the system is running. The formula for the (RE_{BS}) is obtained from the following equation.

$$RE_{BS} \;=\; \frac{\overline{v}_{ref} - \overline{vb}_{(R)}}{\overline{v}_{ref}} \times 100\% \tag{39}$$

where;

$$\overline{v}_{ref} = \sum_{k=1}^{n}\left(v_{reff_{(n)}}\right) n^{-1} \tag{40}$$

$$\overline{vb_R} = \sum_{k=1}^{n}(vb_{R_n})n^{-1} \tag{41}$$

4- Experiment and Result

The experimental testing was divided into two sections to assess the efficacy and feasibility of PID control on each device in controlling the robot movement according to the predetermined trajectory. Section (A) explained the PID tuning in response to any input changes described in Chapter III. The overall PID control was evaluated in section (B) for regulating the robot's speed, the direction of movement, and heading according to the trajectory planning. The final results regarding the performance criteria were analyzed according to the method described in Chapter III to get study conclusions. Figure 8 shows the robot control system; Figure 9 shows the robot architecture system design. Meanwhile, Figure 10 and Table 2 present the robot's parameter and dimension details.

Table 2. Robot and Wheel Parameter Data.

Parameter	Length	Unit
Radius Wheel	50	mm
Circumference (K)	314.28	mm
Distance Wheel to Center of Robot (R)	165	mm
Dimension Robot	400x400	mm
Mass of Robot	3.73	kg
Max Voltage Battery	24	V

4-1- PID Tuning Testing

The trial-and-error method was used to tune each value of K_p, K_i, and K_d for each sub-system on an omnidirectional robot, considering the system response to parameter changes by involving the data in Table 3. The best results were observed and taken in response to plant changes as a result of this process.

4-2- Motor Speed Test

The motor speed was measured at three different speeds in the PID control test: $0.44, 0.6, 0.74$ $({}^{m}/_{s})$. Then, how the PID control reacts to changes in motor speed was observed. The PID tuning method examines the response to every change in the K_p, K_i, and K_d parameter values. The most satisfactory tuning results were obtained after many tuning processes had been completed, as presented in Table 3.

Table 3. Motor Speed PID Tuning Results Parameters.

Parameter	Motor 1	Motor 2	Motor 3	Motor 4
K_p	1.05	1.05	1.05	1.05
K_i	0.05	0.05	0.05	0.05
K_d	0.12	0.12	0.12	0.12
Time Sampling	100ms	100ms	100ms	100ms

Figure 11 shows the system response to the changes of motor speed reference without implementing acceleration control. The diagram shows the speed of each motor: the red-colored line shows the speed of motor 1, the green-colored line shows the speed of motor 2, the blue-colored line shows the speed of motor 3, and the purple-colored line shows the speed of motor 4. Three tests were carried out for each motor, using the set speed points of 0.44, 0.6, and 0.74.

Figure 11 (a) shows the test results for Motor 1. At the set-point speed of 0.44, the overshoot reached 9.31, the biggest overshoot between the test results for all set-points of speed test for Motor 1. Meanwhile, at the set-point speed of 0.6, the overshoot value is 3.21. The smallest overshoot value is 1.62, which was reached by Motor 1 at the referenced speed of 0.74. The average value of overshoot on Motor 1 is 4.71.

Similar results were shown on the speed test results for Motor 2, as shown in Figure 11 (b). The biggest overshoot was reached at the set-point speed of 0.44, which is 7.05. The smallest overshoot, which is 2.17, was gotten at the set-point speed of 0.74. At the set-point speed of 0.6, the overshoot value is 3.21. A similar average of the overshoot was also shown; the average value is 4.14.

Figure 11 (c) shows the speed test results for Motor 3. Although it cannot be seen in the figure, a slight difference compared to the test results of Motor 1 and Motor 2. Unlike in two other previous tests, the smallest overshoot was found at the set-point speed of 0.6, which is 3.21. At the set-point speed of 0.44, the overshoot value is 7.04, which was the biggest. At the set-point speed of 0.74, the overshoot value is 3.52. The average overshoot value for Motor 3 is 4.54.

The speed test results for Motor 4 are shown in Figure 11 (d). In the test results, similar overshoot values were found at the set-point speed of 0.6 and 0.74, which were 4.90 and 4.93, respectively. Meanwhile, at the set-point speed of 0.44, the overshoot value is 7.04. The average overshoot value for Motor 4 is 4.93.

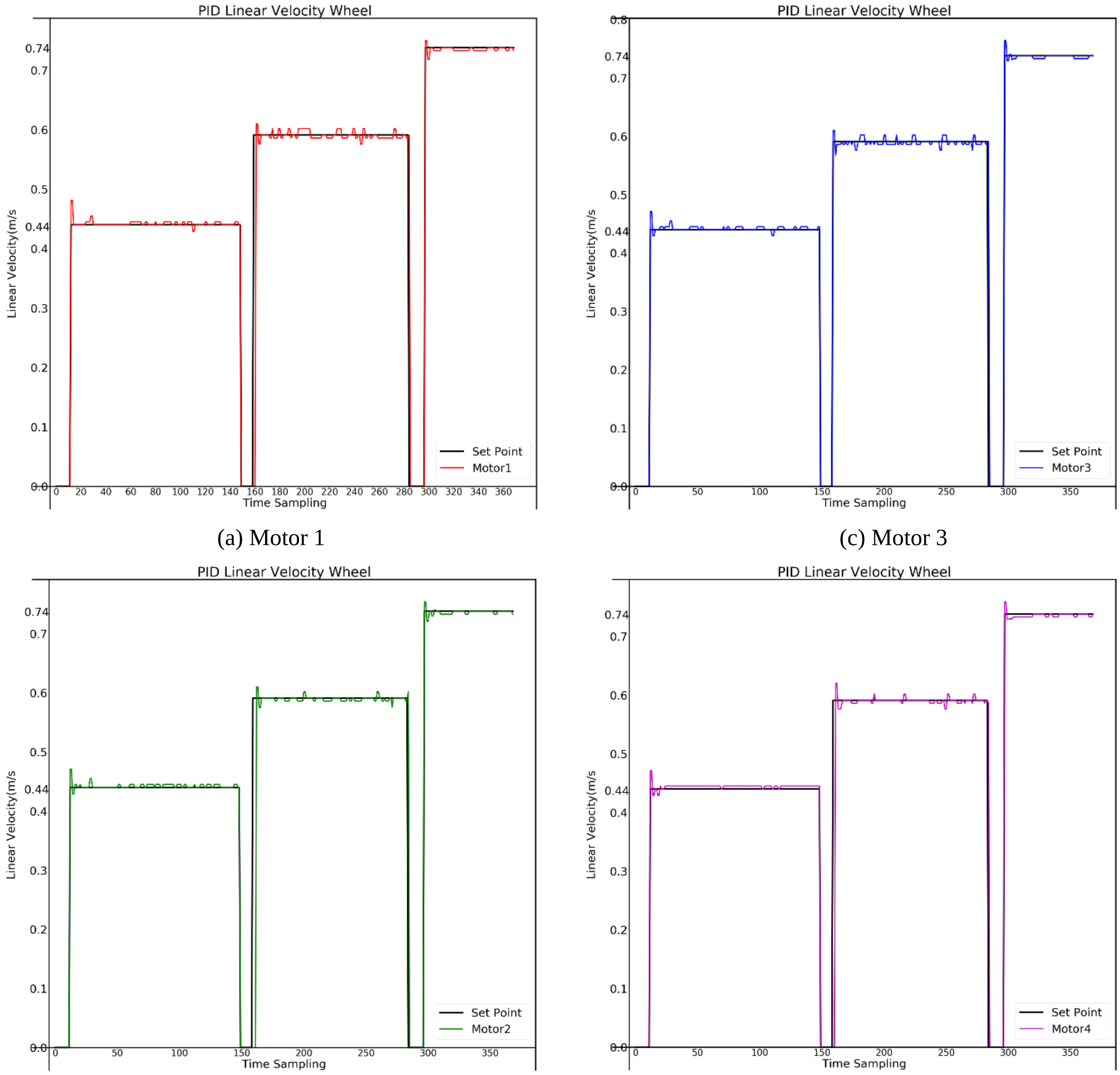
Based on the general speed test results of the four motors, the system performances based on the time response analysis are as follows: rise time is 0.2s, the settling time is 0.5s, and the average maximum overshoot is 5%. These findings showed that the PID control system reacted quickly to changes of input. However, since the system produced an utterly high overshoot value, a significant and sudden change in the motor speed value made the system unsteady, causing the wheels to slip, resulting in an inaccurate odometrical reading.

Figure 12 illustrates the response of motor speed control with acceleration control. The line color representation and the referenced speeds used in this test were the same as the previous one.

The first test was conducted in Motor 1, and the results are shown in Figure 12 (a). Meanwhile, Figure 12 (b), Figure 12 (c), and Figure 12 (d) show the test results for Motor 2, Motor 3, and Motor 4, respectively. Based on the figures, it can be seen that the acceleration control made the wheel system have a blunt shape of linear velocity if compared to the previous test results. The blunt shape of the curve shows the effect of the acceleration control; the wheel tends to move smoother than without the acceleration control. This finding is in line with the overshoot made by the system. The average overshoot for the speed test results with acceleration control is much lower than without the acceleration. The average overshoot value for Motor 1 is 0.89, for Motor 2 is 0.67, for Motor 3 is 0.83, and for Motor 4 is 0.53.

Other proof of smoother response is also found based on the time-response performance parameters in general. The findings show that the system responded without overshoot at the start of the system. The system also responded slower than in the previous test with 0.8s rise time and 1s settling time, which typically happens with the smooth movement of the wheel.

More detailed test results of overshoot for each motor at every set-point speed will be mentioned as follows. In the responses of Motor 1, at the set-point speed of 0.44, the overshoot is 1.59. Meanwhile, at the set-point speed of 0.6, the overshoot is 0.68; and at the speed of 0.74, the overshoot is 0.41. According to speed test results on Motor 2, the overshoot is 1.14 at the set-point speed of 0.44. The overshoot is 0.34 at the set-point speed of 0.6 and 0.54 at the set-point speed of 0.74. Meanwhile, Motor 3 has a 1.14 overshoot at the set-point speed of 0.44. Motor 3 has a 0.68 overshoot at the et-point speed of 0.6 and a 0.68 overshoot at the set-point speed of 0.74. Lastly, in the responses of Motor 4, at the set-point speed of 0.44, the overshoot is 1.14. At the set-point speed of 0.6, Motor 4 has a 0.17 overshoot. At the set-point speed of 0.74, it has a 0.27 overshoot.



(a) Motor 1

(c) Motor 3

(b) Motor 2

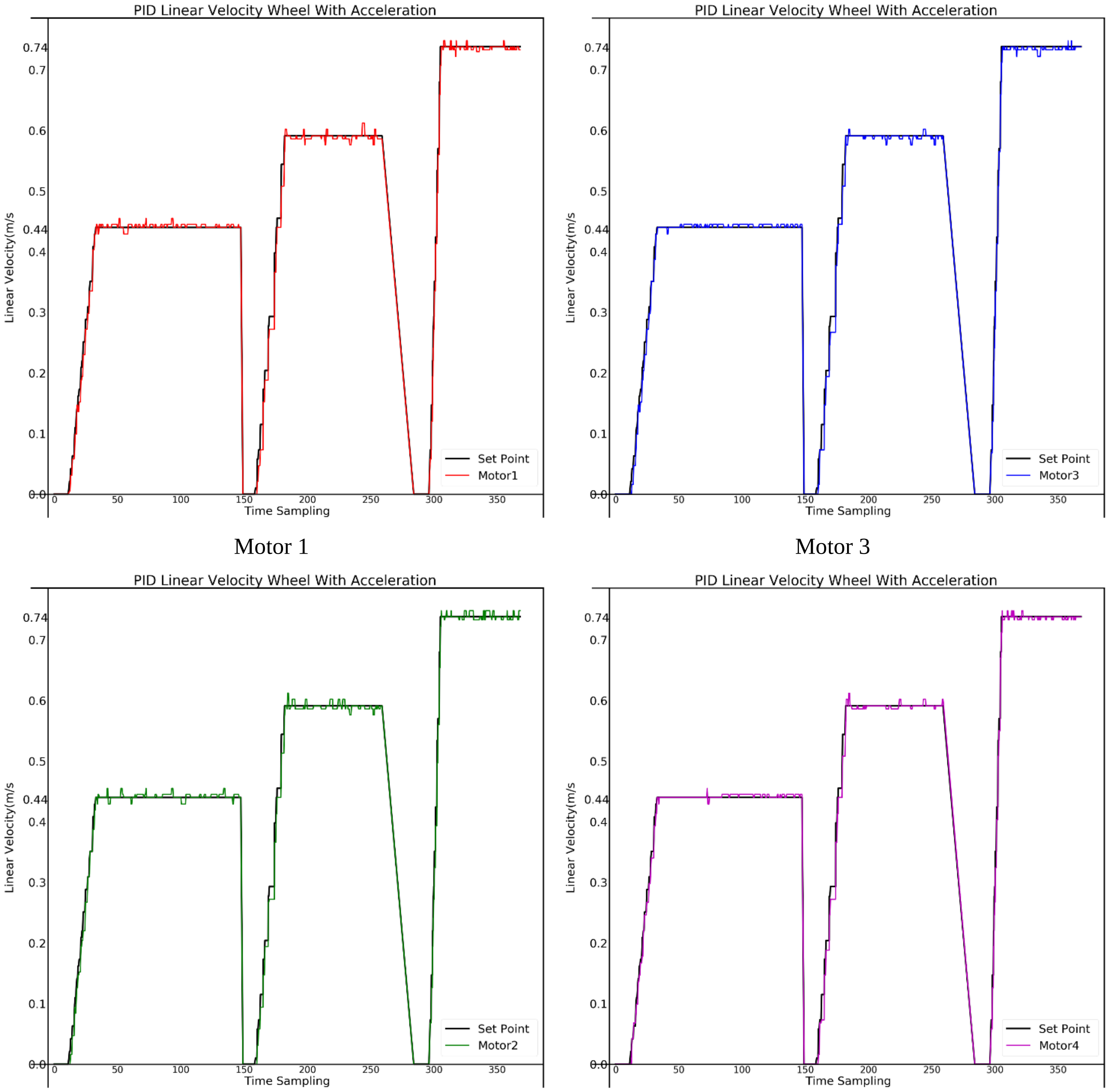
(d) Motor 4

Figure 11. Response of Motor Speed Control to Various Set-points without Acceleration Control.

4-3-Robot Motion Test

The determination of K_p, K_i, and K_d for PID Motion Control was carried out using the trial-and-error method, and the parameters of the best result are presented in Table 4. The test results are presented in system responses of environmental changes were observed. Three tests were carried out in this test. All of the tests were begun with initialization, which included three positions: robot starting position, an initial position of trajectory planning, and final position of trajectory planning. The initial and final positions of trajectory planning were fixed in (0.0, 0.0) and (0.0, 4.5) in meter units, respectively.

Meanwhile, robot starting positions were different for each test. Test results were observed and presented in graphs as in Figures 13 to 15. The red color in the graph represents the robot start position to the trajectory planning; the blue color represents the initial position of trajectory planning; the yellow color represents the end of the trajectory.



Motor 1

Motor 3

Motor 2

Motor 4

Figure 12. Response of Motor Speed Control to Set-points with Acceleration.

Table 4. Parameters of PID Motion Tuning Results.

Parameter	**Value**
K_p	3.5
K_i	0.42
K_d	0.15

The first test was performed with the robot placed at (0.5, -1.5) in the meter unit. The robot moves from this position to the end position of the trajectory planning, as shown with the green-colored line in Figure 13. In the first test, the robot began to follow the trajectory at (0.0, 2.5).

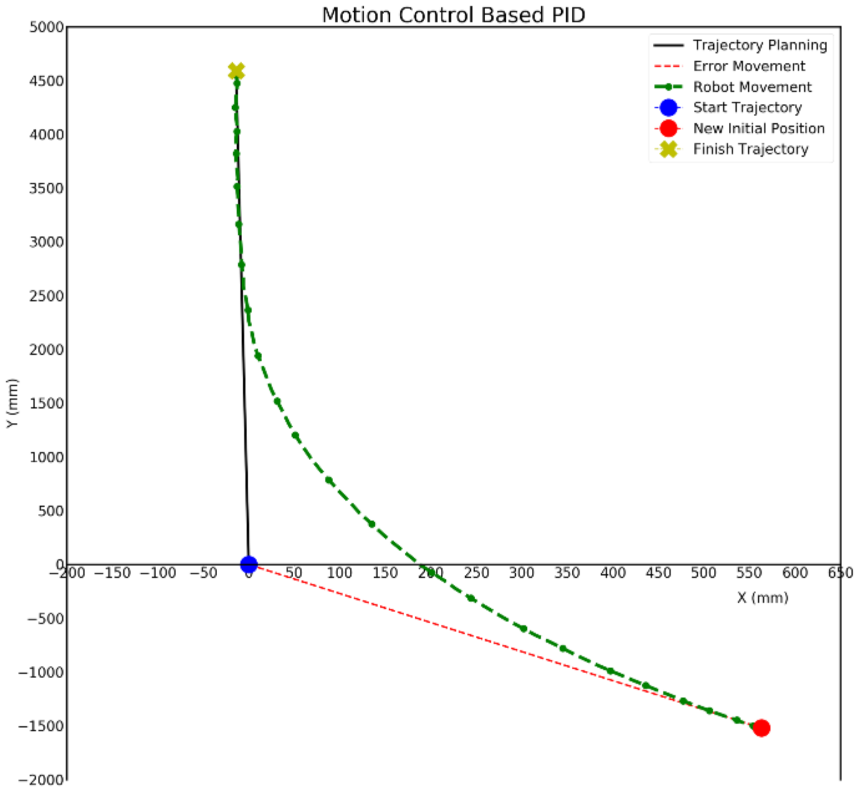


Figure 13. System Response of PID Motion Control in Coordinates of the Experimental Robot: Movement 1.

In the second test, the robot was placed at (-2.5, -4.5) meters. The test result is presented in Figure 14; the robot began to follow the trajectory planning at (0.0, 2.5) meters.

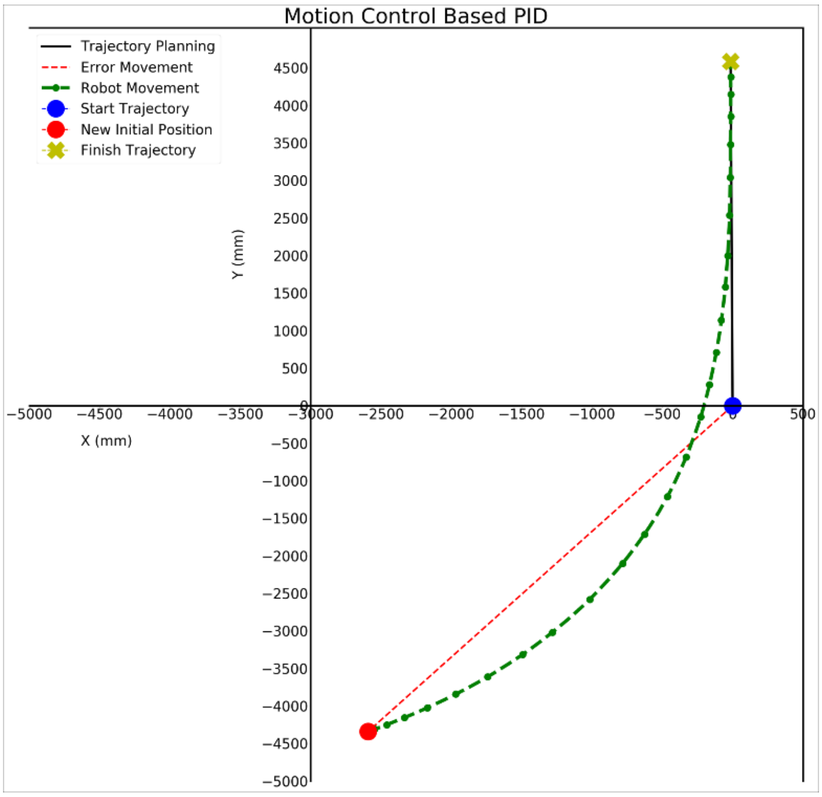


Figure 14. System Response of PID Motion Control in Coordinates of the Experimental Robot: Movement 2.

The third test was performed with the robot placed at (4.5m, -4.5m). Based on the test result in Figure 15, the robot began to follow the trajectory at (0.0, 2.0) meters.

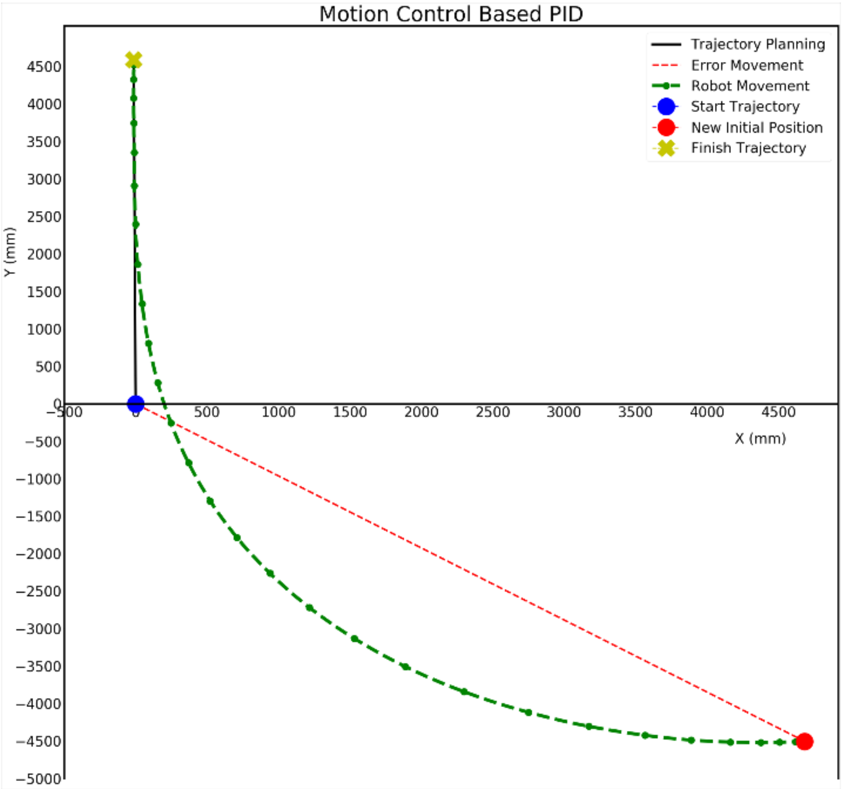


Figure 15. System Response of PID Motion Control in Coordinates of the Experimental Robot: Movement 3.

All motion test results show that the robot could reach the finish position. However, the robot motion test results also show that the robot did not track the trajectory planning from the initial position of trajectory planning; it took some time after the robot reached the initial position to track the trajectory. Nevertheless, the system responded fastly during this time. The robot motion test results are also similar to an exponential response; it should be noted that the robot did not surpass the trajectory line. It can be seen that the system response has exponential-like response characteristics, no overshoot, and fast rise and settling time.

4-4- Robot Heading

The PID control for robot heading was evaluated by changing the PID input, which is the robot heading angle position, and then observing the system response. The PID tuning process was performed using the same tuning method as the Robot Motion Control Test. The best result of the tuning is presented in Table 5.

Table 5. PID Heading Tuning Results Parameters.

Parameter	Value
K_p	3.0
K_i	0.15
K_d	0.23
Time Sampling	100ms

Figure 16 displays the system response with PID-based heading control for changes of heading angle. According to the graph in Figure 16, the robot heading angles can track the given set-points without any issue. The red-colored line, which is the actual robot heading, was coincided with the black-colored line, which is the referenced heading angle.

More detailed observational data will be explained as follows. It shows that the overshoot is 1.428 for 350° angle, 0 for 30°, 0 for 90°, 1.67 for 180°, 0 for 230 °, 0 for 350 °, 0 for 0°, and 10 for 180°. The settling time is 10ms for 350° angle, 0ms for 90°, 60ms for 180 °, 0ms for 230°, 0ms for 350 °, 0ms for 0°, and 60ms for 180°. The rise time is 20ms for 350 ° angle, 100ms for 30°, 140ms for 90°, 160ms for 180°, 110ms for 230°, 140ms for 350°, 20ms for 10°, and 150ms for 180°. Based on those results, overall, the system responded with performance specifications: ± 0.4s rise time, 0.8s settling time, and the 2% maximum overshoot. It shows that PID-based heading control can control the robot heading with fast response and small steady-state error and oscillations. Thus, the heading control system is said to be a fairly stable control.

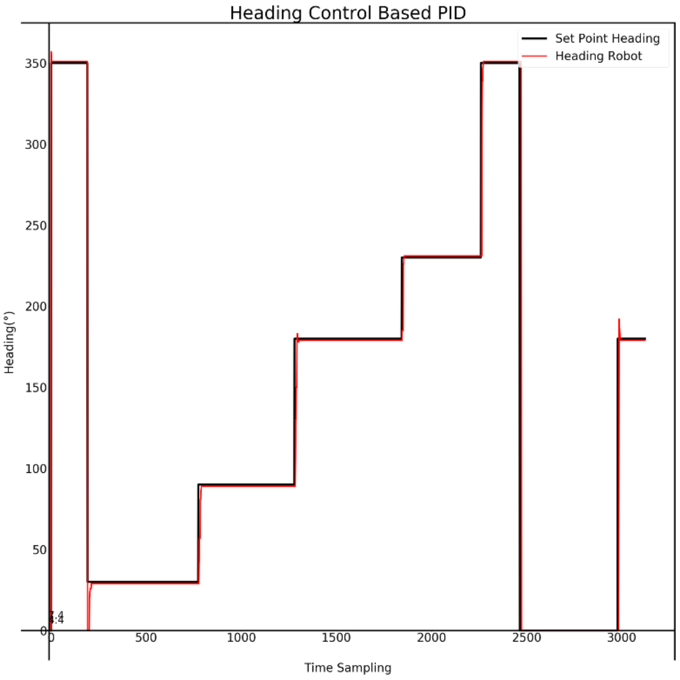


Figure 16. PID-based Heading Control System Response to Input Changes.

4-5- Trajectory Tracking Test

In previous tests, each of the three proposed PID-based controls for trajectory tracking control of the Covid-19 aromatherapy robot is tested and evaluated individually. However, evaluation for the combination of the three proposed PID-based control has not been done. This test was conducted to evaluate the effectiveness and efficiency of the three proposed PID-based controls for trajectory tracking control of the robot; to re-check if using three of them together is necessary or not.

The first test was performed by testing the PID-based controls for three control scenarios: PID 1, which consists of the motor speed, heading, and motion control; PID 2, which consists of the motor speed and motion control; and PID 3, which is only the motor speed control. These control scenarios were tested for trajectory tracking with a triangular-shaped model.

In this test, the graph of the linear base robot velocity $(vb_{(sp)})$ of 0.41 $^{m}/_{s}$ was measured using the three scenarios on a triangle tracking in an unknown environment, as shown in Figure 17. The robot was placed at (0.0; 0.0) and moved to the right to follow the triangular-shaped trajectory planning. The red-colored line showed the first scenario; the yellow-colored line showed the second; the blue-colored line showed the third. The green-colored line represented the planned trajectory. Based on the figure, it can be seen that there are slight tracking errors from the blue-colored line, which was out of the trajectory. Meanwhile, PID 1 and PID 2 had no problem tracking the planned trajectory.

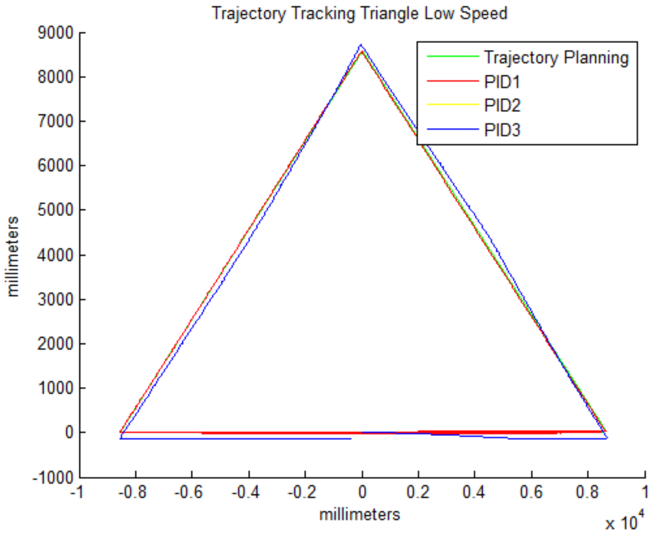


Figure 17. Trajectory Tracking of Three PID Scenarios with Triangular Model.

Then, the robot movement direction of the three control scenarios while tracking the triangular-shaped trajectory model was also evaluated. Figure 18 shows the test results. The color representation used in this test is different from the previous one. PID 1 is represented with the yellow color; PID 2 with the green color; PID 3 with the red color; the set-point with the blue color. According to the figure, the green-colored and yellow-colored lines coincide with the blue-colored line. This means that robots with PID 1 and PID 2 control scenarios tracked the desired direction without any issue. However, the red-colored line could not follow the blue-colored line; PID 3 could not track the referenced direction.

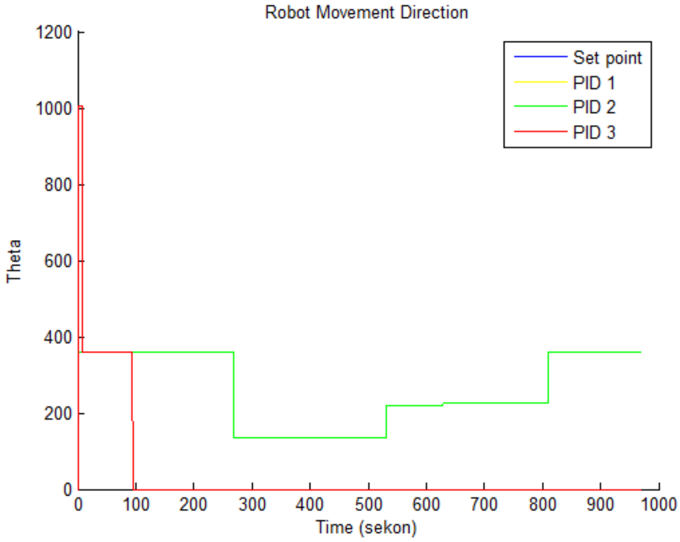


Figure 18. Robot Movement Direction of Three PID Scenarios.

The robot movement orientation of the three control scenarios while tracking the triangular-shaped trajectory model was also evaluated. The test results are represented in Figure 19. The green-colored line has the same value as the blue-colored line, which is zero. The color representation used in this test is different from the previous one. PID 1 is represented with the green color; PID 2 with the blue color; PID 3 with the red color; the set-point with the black color. It can be seen that PID 1 coincided with the set-point while the others did not coincide with the set-point.

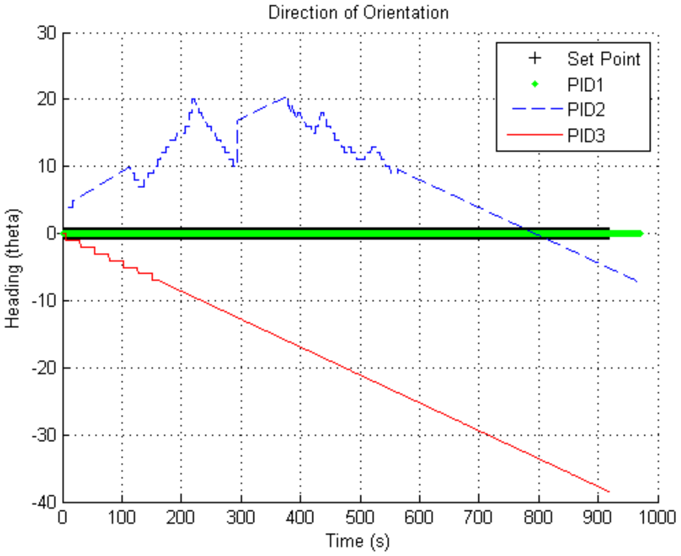


Figure 19. Robot Movement Orientation of Three PID Scenarios.

The second test was performed to test the performance of PID-based controls on trajectory tracking, which used different trajectory models. The scenarios to determine the PID response to trajectory planning were carried out three times with different variations in the linear base robot speed values: low, medium, and high speeds (0.41; 0.83; 1.25 m⁄s). The result will also be analyzed using several performance criteria formulas with IAE_H, IAE_{xy}, MAE_H, MAE_{xy}, $RMSE_H$, $RMSE_{xy}$, and RE_{BS}.

The speed, movement, and PID heading controls on a star-shaped trajectory tracking performance in an unknown environment were measured, as shown in Figure 20. Three-speed tests were performed on the trajectory tracking star: 0.41 m⁄s shown with the blue-colored line, 0.83 m/s shown with a yellow-colored line, and 1.25m⁄s shown with a red-colored line. The robot is placed at the initial position of (0.0, 0.0) and moved to follow the trajectory tracking. It stops at the position of (2.4, 5.8) meters. The findings showed that using the three PID controls can make the robot remain on the trajectory tracking with referenced speeds of 0.41, 0.83, and 1.25 m/s.

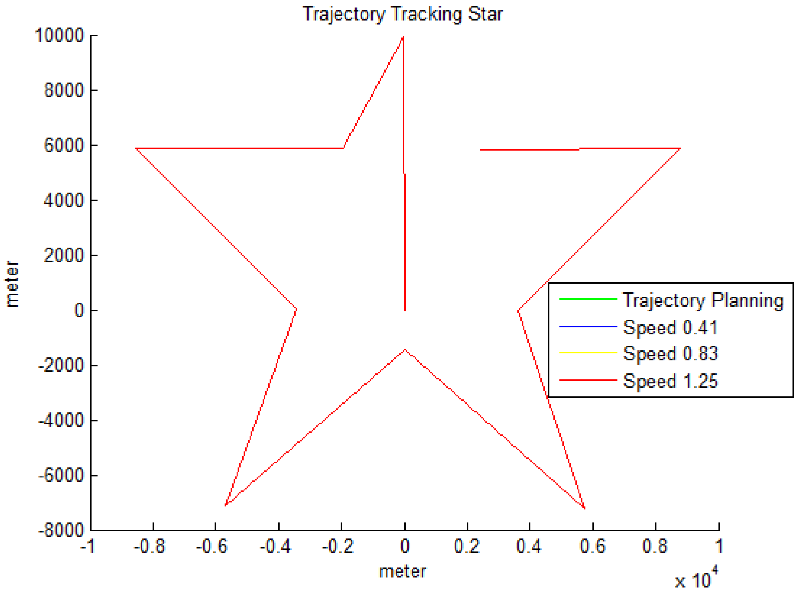


Figure 20. Trajectory tracking with star model.

Then, the error heading robot was observed. The error lines between the system response with three referenced speeds (low, medium, high) are shown in Figure 21. The error between low-speed references with the system response is represented in the green-colored line. Meanwhile, the red-colored line represents the error of medium-speed, and the blue-colored line represents the error of high-speed reference. Based on the figure, it can be seen that the blue-colored line contains heading errors, but not with the red-colored and green-colored lines.

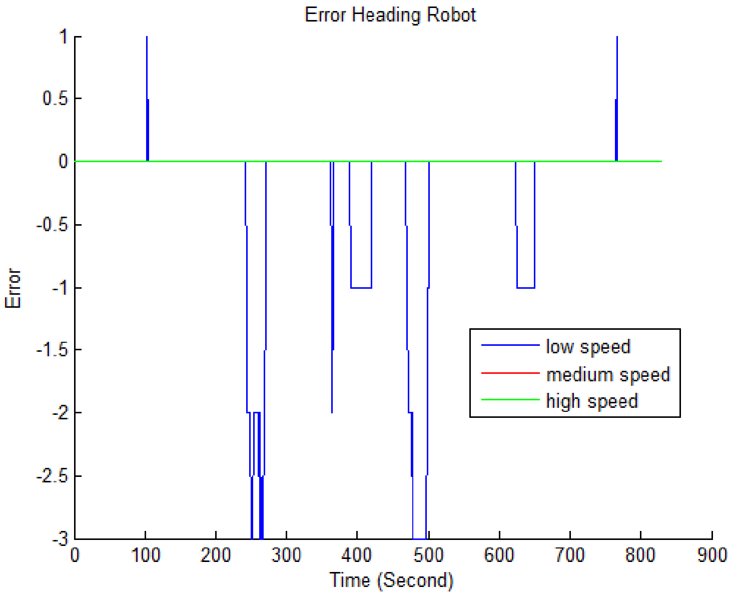


Figure 21. Heading Robot Error.

The robot's linear velocity was also observed. The trajectory tracking performance was measured with a low speed of 0.4187 $^{m}/_{s}$. The blue-colored line in Figure 22 shows the set-point velocity. The red color represents the robot velocity, and the green color represents the error between robot velocity and its set-point. The findings show that the maximum value of the robot speed on the red line curve is 0.4239 $^{m}/_{s}$. The largest error value shown is 0.4125 $^{m}/_{s}$.

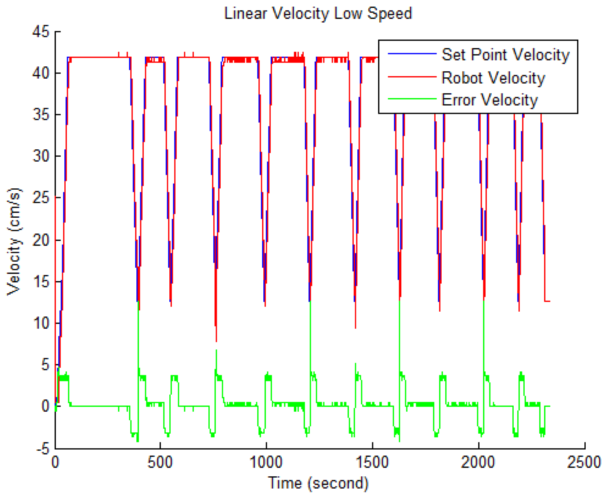


Figure 22. **Low-speed Performance in Linear Velocity.**

Then, the trajectory tracking performance of medium speed of 0.8373 $^m/_s$ is represented in Figure 23. According to the results, the maximum value of the robot speed is 0.8426 $^m/_s$. The largest error value is 0.1465 $^m/_s$.

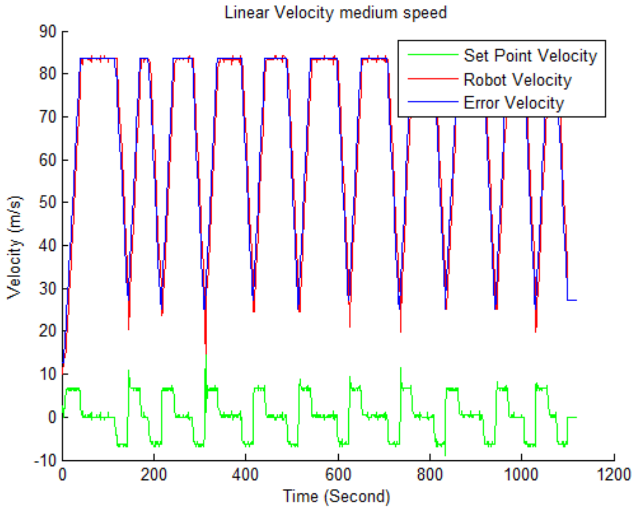


Figure 23. **Medium-speed Performance in Linear Velocity.**

The performance of the robot in high speed (1.256 m/s) is shown in Figure 24. The maximum value of the robot speed is 0.8426 $^m/_s$, while the largest error is 0.1465 $^m/_s$.

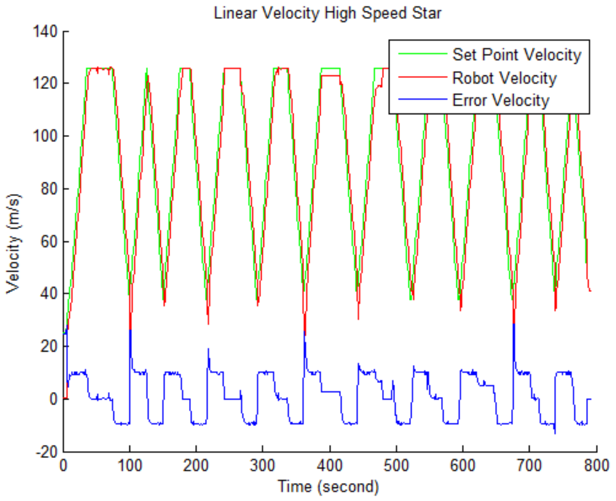


Figure 24. **High-speed Performance in Linear Velocity.**

Based on the test results regarding the robot's linear velocity, the robot can consistently track the trajectory with minor errors and oscillations. A stable linear velocity means the robot can move stably and consistently without jolts or sudden movements. This also means that the Covid-19 aromatherapy robot can perform well in low-speed, medium-speed, and even high-speed, making it have fast travel time with stable speed movement.

Meanwhile, the robot's position control performance in low-speed, medium-speed, and high-speed configurations can be seen in Figure 25, Figure 26, and Figure 27, respectively. The robot position and the referenced position while the robot was tracking a star-model trajectory were recorded. The difference between the actual and the referenced position was calculated as an error.

The error on the X-axis position is shown in the pink-colored line, and the error on the Y-axis position is shown in the light-blue-colored line. Meanwhile, the red-colored line represents the set-point position on the X-axis, and the blue-colored line represents the Y-axis. The green color represents the robot's actual position on X-axis, yellow represents the robot position on Y-axis.

Position control performance for low-speed configuration is shown in Figure 25, which can be seen that the robot position almost coincides with the set-point. The error found is less than 1 cm.

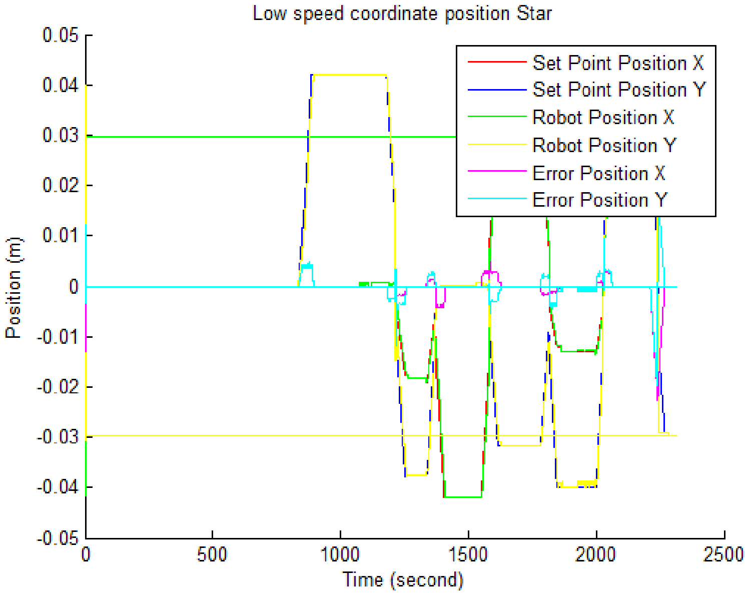


Figure 25. Position coordinates for low speed.

Meanwhile, position control performance for medium-speed configuration is shown in Figure 26. The results were found similar to low-speed configuration results, which have minimal errors of positions.

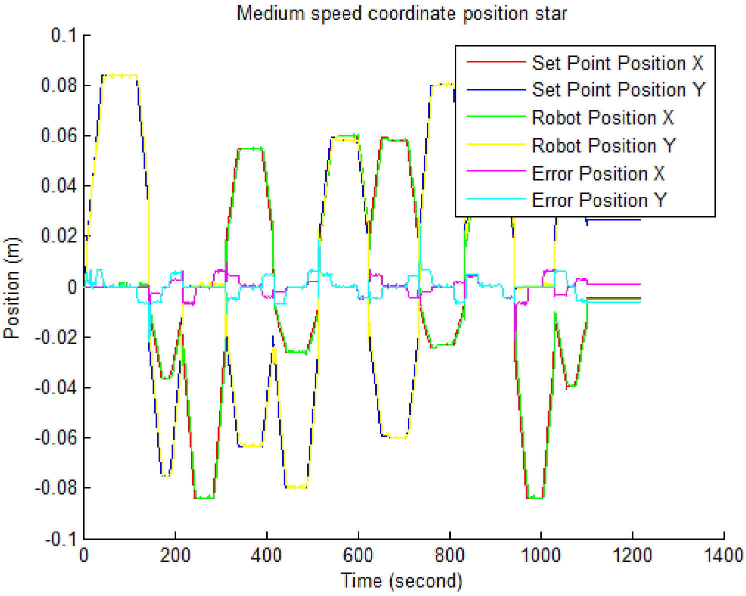


Figure 26. Position coordinates for medium speed.

Alike in test results for low and medium speed configurations, the test results for high-speed configuration, as in Figure 27, have minimal errors. However, the position errors for high-speed configuration were a bit larger than previous test results, although it still had the result close to zero.

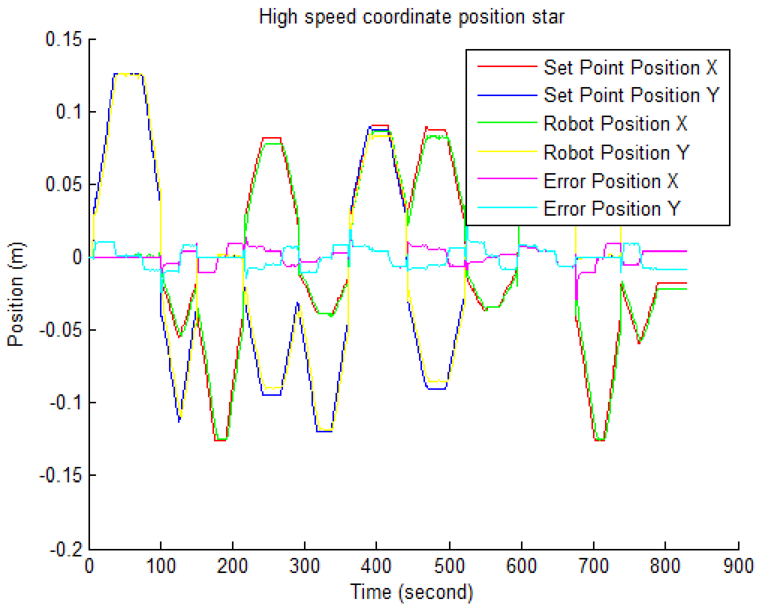
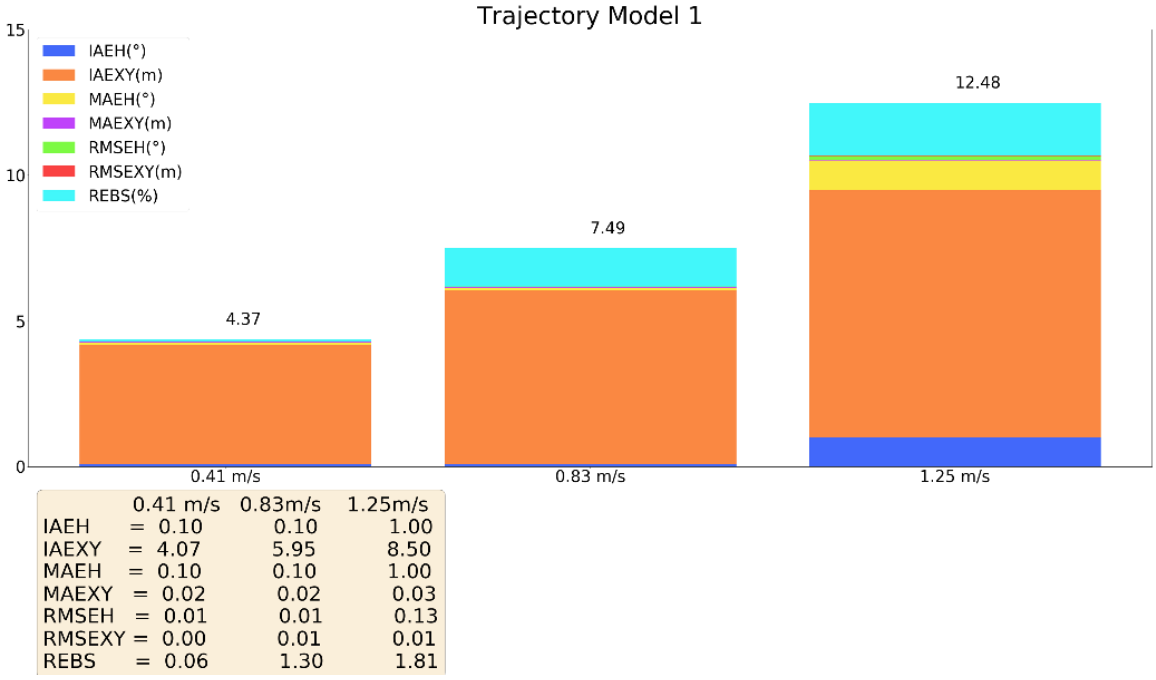


Figure 27. Position coordinated for high speed.

Finally, the proposed PID-based trajectory tracking control for Covid-19 aromatherapy robot were evaluated based on the evaluation parameters. Figure 28 shows seven parameters: IAEH shown in the blue graph, IAEXY in the orange, MAEH shown in the yellow, MAEXY in the purple, RMSEH in the green, RMSEXY in the red, and REBS in the cyan. Based on the performance criteria of the graphic image, the IAEH performance value at low speed is 0.1, at medium speed is 0.1, and at high speed is 1. The IAEXY performance value at low speed is 4.0, at medium is 5, and at high is 8.50. The MAEH shows the low-performance value of 0.1 at low speed and medium speed and demonstrates the highest performance value of 1 at high speed. The highest performance value of the RMSEH is 0.13 at high speed and the lowest value 0.01 at low and medium speeds. RMSEXY performance shows the highest value at high and medium speeds, while at low speed, it shows a low-performance value of 0.00. The REBS performance value for low speed is 0.06, for medium speed is 1.30, and for high speed, it is 1.81.



	0.41 m/s	0.83m/s	1.25m/s
IAEH =	0.10	0.10	1.00
IAEXY =	4.07	5.95	8.50
MAEH =	0.10	0.10	1.00
MAEXY =	0.02	0.02	0.03
RMSEH =	0.01	0.01	0.13
RMSEXY =	0.00	0.01	0.01
REBS =	0.06	1.30	1.81

Figure 28. Seven parameters performance robot.

5- Conclusion

In this study, a four-wheel omnidirectional robot's trajectory tracking control method was successfully developed specifically for the Covid-19 aromatherapy robot. Errors in setting the speed, robot movement direction, and changes in the robot orientation can all cause robot movement errors in a trajectory. The PID control is used to control three key parameters in this study: motor speed, robot movement direction, and robot orientation direction. The PID and acceleration systems are combined using the odometry system as feedback from the trajectory tracking, allowing the robot to operate precisely and steadily. According to the study results, the error rate of the robot's linear velocity is 2%, with an average test value of 1.3%. The average robot's movement error at coordinates X and Y is 0.0059 m. The robot heading error on all trajectories is 0.6%. The robot can be controlled and maintained to stay on the trajectory tracking. It is expected that further research will be carried out by adding an IMU sensor to the robot design. This improvement aims to detect robot movement. It can be combined with an odometrical sensor to detect movement disturbances such as wheel slip, modeling kinematics errors due to improper robot construction errors. It is also expected to improve the robot's inhalation delivery mechanism so that it can also be improved to be a nebulizer robot.

6- Declarations

6-1- Author Contributions

Conceptualization, I.; methodology, N.M.R; hardware and software, D.U.R.; validation, R.S.; resources, F.A.; writing—original draft preparation, I.; writing—review and editing, A.M.; supervision, G.S. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data presented in this study are available in article.

6-3- Funding

The authors received financial support for the research of this article from Universitas Muhammadiyah Yogyakarta.

6-4- Acknowledgements

The authors would like to thank Universitas Muhammadiyah Yogyakarta for supporting this work by providing a research grant for this study.

6-5- Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Banik, Jozef, Vojtech Mezera, Christian Köhler, and Marco Schmidtmann. “Antiplatelet Therapy in Patients with Covid-19: A Retrospective Observational Study.” Thrombosis Update 2 (2021): 100026. doi:10.1016/j.tru.2020.100026.

[2] Godino, Cosmo, Andrea Scotti, Norma Maugeri, Nicasio Mancini, Evgeny Fominskiy, Alberto Margonato, and Giovanni Landoni. “Antithrombotic Therapy in Patients with COVID-19? -Rationale and Evidence.” International Journal of Cardiology 324 (February 2021): 261–266. doi:10.1016/j.ijcard.2020.09.064.

[3] Russo, Vincenzo, Marco Di Maio, Emilio Attena, Angelo Silverio, Fernando Scudiero, Dario Celentani, Corrado Lodigiani, and Pierpaolo Di Micco. “Clinical Impact of Pre-Admission Antithrombotic Therapy in Hospitalized Patients with COVID-19: A Multicenter Observational Study.” Pharmacological Research 159 (September 2020): 104965. doi:10.1016/j.phrs.2020.104965.

[4] Russo, Vincenzo, Marco Di Maio, Emilio Attena, Angelo Silverio, Fernando Scudiero, Dario Celentani, Corrado Lodigiani, and Pierpaolo Di Micco. “Clinical Impact of Pre-Admission Antithrombotic Therapy in Hospitalized Patients with COVID-19: A Multicenter Observational Study.” Pharmacological Research 159 (n.d.): 104965. doi:10.1016/j.phrs.2020.104965.

[5] Formisano, Elena, Pasquale Di Maio, Cecilia Ivaldi, Elsa Sferrazzo, Lorenzina Arieta, Silvia Bongiovanni, and Loredana Panizzi. “Nutritional Therapy for Patients with Coronavirus Disease 2019 (COVID-19): Practical Protocol from a Single Center Highly Affected by an Outbreak of the Novel Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) Infection.” Nutrition 82 (February 2021): 111048. doi:10.1016/j.nut.2020.111048.

[6] Altuntas, Fevzi, Naim Ata, Tugce Nur Yigenoglu, Semih Bascı, Mehmet Sinan Dal, Serdal Korkmaz, and Sinem Namdaroglu. “Convalescent Plasma Therapy in Patients with COVID-19.” Transfusion and Apheresis Science, no. August (September 2020): 102955. doi:10.1016/j.transci.2020.102955.

[7] Ghasemiyeh, Parisa, and Soliman Mohammadi-Samani. “COVID-19 Outbreak: Challenges in Pharmacotherapy Based on Pharmacokinetic and Pharmacodynamic Aspects of Drug Therapy in Patients with Moderate to Severe Infection.” Heart & Lung 49, no. 6 (2020): 763–73. doi:10.1016/j.hrtlng.2020.08.025.

[8] Kaminski, Monica A, Subin Sunny, Khayala Balabayova, Avneet Kaur, Aanchal Gupta, Marie Abdallah, and John Quale. “Tocilizumab Therapy for COVID-19: A Comparison of Subcutaneous and Intravenous Therapies.” International Journal of Infectious Diseases 101 (December 2020): 59–64. doi:10.1016/j.ijid.2020.09.1447.

[9] Gowrishankar, Shanmugaraj, Sankar Muthumanickam, Arumugam Kamaladevi, Chandrasekar Karthika, Ravi Jothi, Pandi Boomi, Dharuman Maniazhagu, and Shunmugiah Karutha Pandian. “Promising Phytochemicals of Traditional Indian Herbal Steam Inhalation Therapy to Combat COVID-19 – An in Silico Study.” Food and Chemical Toxicology 148 (September 2020): 111966. doi:10.1016/j.fct.2020.111966.

[10] Shah, Mili, Jignasha Captain, Vidyadhar Vaidya, Arvind Kulkarni, Kedar Valsangkar, Pradeep M K Nair, and Gayatri Ganu. “Safety and Efficacy of Ozone Therapy in Mild to Moderate COVID-19 Patients: A Phase 1/11 Randomized Control Trial (SEOT Study.” International Immunopharmacology 91 (February 2021): 107301. doi:10.1016/j.intimp.2020.107301.

[11] Eedara, B B, W Alabsi, D Encinas-Basurto, R Polt, J G Ledford, and H M Mansour. “Inhalation Delivery for the Treatment and Prevention of COVID-19 Infection.” Pharmaceutics 13, no. 7 (2021): 1077. doi:10.3390/pharmaceutics13071077.

[12] Yan, Fanlei, Baoquan Li, Wuxi Shi, and Dongwei Wang. “Hybrid Visual Servo Trajectory Tracking of Wheeled Mobile Robots.” IEEE Access 6 (2018): 24291–24298. doi:10.1109/ACCESS.2018.2829839.

[13] Yang, Hongjiu, Mingchao Guo, Yuanqing Xia, and Lei Cheng. “Trajectory Tracking for Wheeled Mobile Robots via Model Predictive Control with Softening Constraints.” IET Control Theory & Applications 12, no. 2 (January 2018): 206–214. doi:10.1049/iet-cta.2017.0395.

[14] Chang, Hongbin, Shuoyu Wang, and Ping Sun. “Omniwheel Touchdown Characteristics and Adaptive Saturated Control for a Human Support Robot.” IEEE Access 6 (2018): 51174–51186. doi:10.1109/ACCESS.2018.2869836.

[15] Li, Zhijun, Xiaoqing Cao, Yong Tang, Rui Li, and Wenjun Ye. “Bilateral Teleoperation of Holonomic Constrained Robotic Systems with Time-Varying Delays.” IEEE Transactions on Instrumentation and Measurement 62, no. 4 (2013): 752–765. doi:10.1109/TIM.2013.2246906.

[16] Huang, Hsu-Chih. “SoPC-Based Parallel ACO Algorithm and Its Application to Optimal Motion Controller Design for Intelligent Omnidirectional Mobile Robots.” IEEE Transactions on Industrial Informatics 9, no. 4 (2013): 1828–1835. doi:10.1109/TII.2012.2222033.

[17] Rotondo, Damiano, Vicenc Puig, Fatiha Nejjari, and Juli Romera. “A Fault-Hiding Approach for the Switching Quasi-LPV Fault Tolerant Control of a Four-Wheeled Omnidirectional Mobile Robot.” IEEE Transactions on Industrial Electronics 62, no. 6 (2015): 1–1. doi:10.1109/TIE.2014.2367002.

[18] Lee, Ming-Han, and Tzuu-Hseng S Li. “Kinematics, Dynamics and Control Design of 4WIS4WID Mobile Robots.” The Journal of Engineering, no. 1 (2015): 6–16. doi:10.1049/joe.2014.0241.

[19] Bresson, Guillaume, Thomas Feraud, Romuald Aufrere, Paul Checchin, and Roland Chapuis. “Real-Time Monocular SLAM With Low Memory Requirements.” IEEE Transactions on Intelligent Transportation Systems 16, no. 4 (2015): 1827–1839. doi:10.1109/TITS.2014.2376780.

[20] Sanchez, C M, J R G Sanchez, C Y S Cervantes, R S Ortigoza, V M H Guzman, J N A Juarez, and M M Aranda. “Trajectory Generation for Wheeled Mobile Robots Via Bézier Polynomials.” IEEE Latin America Transactions 14, no. 11 (2016): 4482–4490. doi:10.1109/TLA.2016.7795818.

[21] Li, Luyang, Yun-Hui Liu, Tianjiao Jiang, Kai Wang, and Mu Fang. “Adaptive Trajectory Tracking of Nonholonomic Mobile Robots Using Vision-Based Position and Velocity Estimation.” IEEE Transactions on Cybernetics 48, no. 2 (2018): 571–582. doi:10.1109/TCYB.2016.2646719.

[22] Pereida, Karime, Mohamed K Helwa, and Angela P Schoellig. “Data-Efficient Multirobot, Multitask Transfer Learning for Trajectory Tracking.” IEEE Robotics and Automation Letters 3, no. 2 (2018): 1260–1267. doi:10.1109/LRA.2018.2795653.

[23] Saradagi, Akshit, Vijay Muralidharan, Vishaal Krishnan, Sandeep Menta, and Arun D Mahindrakar. “Formation Control and Trajectory Tracking of Nonholonomic Mobile Robots.” IEEE Transactions on Control Systems Technology 26, no. 6 (2018): 2250–2258. doi:10.1109/TCST.2017.2749563.

[24] Santos, Milton Cesar Paes, Claudio Dario Rosales, Mario Sarcinelli-Filho, and Ricardo Carelli. “A Novel Null-Space-Based UAV Trajectory Tracking Controller with Collision Avoidance.” IEEE/ASME Transactions on Mechatronics 22, no. 6 (2017): 2543–2553. doi:10.1109/TMECH.2017.2752302.

[25] He, Wei, Tingting Meng, Shuang Zhang, Quanbo Ge, and Changyin Sun. “Trajectory Tracking Control for the Flexible Wings of a Micro Aerial Vehicle.” IEEE Transactions on Systems, Man, and Cybernetics: Systems 48, no. 12 (2018): 2431–2441. doi:10.1109/TSMC.2017.2779854.

[26] Kayacan, Erdal, and Reinaldo Maslim. “Type-2 Fuzzy Logic Trajectory Tracking Control of Quadrotor VTOL Aircraft with Elliptic Membership Functions.” IEEE/ASME Transactions on Mechatronics 22, no. 1 (2017): 339–348. doi:10.1109/TMECH.2016.2614672.

[27] Antonelli, Gianluca, Elisabetta Cataldi, Filippo Arrichiello, Paolo Robuffo Giordano, Stefano Chiaverini, and Antonio Franchi. “Adaptive Trajectory Tracking for Quadrotor MAVs in Presence of Parameter Uncertainties and External Disturbances.” IEEE Transactions on Control Systems Technology 26, no. 1 (2018): 248–254. doi:10.1109/TCST.2017.2650679.

[28] Baek, Jaemin, Soonwan Cho, and Soohee Han. “Practical Time-Delay Control with Adaptive Gains for Trajectory Tracking of Robot Manipulators.” IEEE Transactions on Industrial Electronics 65, no. 7 (2018): 5682–5692. doi:10.1109/TIE.2017.2782238.

[29] Yang, Lidong, Qianqian Wang, and Li Zhang. “Model-Free Trajectory Tracking Control of Two-Particle Magnetic Microrobot.” IEEE Transactions on Nanotechnology 17, no. 4 (2018): 697–700. doi:10.1109/TNANO.2018.2815978.

[30] Angelini, Franco, Cosimo Della Santina, Manolo Garabini, Matteo Bianchi, Gian Maria Gasparri, Giorgio Grioli, Manuel Giuseppe Catalano, and Antonio Bicchi. “Decentralized Trajectory Tracking Control for Soft Robots Interacting With the Environment.” IEEE Transactions on Robotics 34, no. 4 (2018): 924–935. doi:10.1109/TRO.2018.2830351.

[31] Wang, Zerui, Sing Chun Lee, Fangxun Zhong, David Navarro-Alarcon, Yun Hui Liu, Anton Deguet, Peter Kazanzides, and Russell H Taylor. “Image-Based Trajectory Tracking Control of 4-DoF Laparoscopic Instruments Using a Rotation Distinguishing Marker.” IEEE Robotics and Automation Letters 2, no. 3 (2017): 1586–1592. doi:10.1109/LRA.2017.2676350.

[32] Chen, Jian, Bingxi Jia, and Kaixiang Zhang. “Trifocal Tensor-Based Adaptive Visual Trajectory Tracking Control of Mobile Robots.” IEEE Transactions on Cybernetics 47, no. 11 (November 2017): 3784–3798. doi:10.1109/TCYB.2016.2582210.

[33] Ashrafiuon, Hashem, Sergey Nersesov, and Garrett Clayton. “Trajectory Tracking Control of Planar Underactuated Vehicles.” IEEE Transactions on Automatic Control 62, no. 4 (2017). doi:10.1109/TAC.2016.2584180.

[34] Yu, Xiao, and Lu Liu. “Target Enclosing and Trajectory Tracking for a Mobile Robot with Input Disturbances.” IEEE Control Systems Letters 1, no. 2 (2017): 221–226. doi:10.1109/LCSYS.2017.2712663.

[35] Sakthivel, R, S Harshavarthini, R Kavikumar, and Yong-Ki Ma. “Robust Tracking Control for Fuzzy Markovian Jump Systems with Time-Varying Delay and Disturbances.” IEEE Access 6 (2018): 66861–66869. doi:10.1109/ACCESS.2018.2879379.

[36] Yuan, Mingxing, Zheng Chen, Bin Yao, and Jinfei Hu. “An Improved Online Trajectory Planner With Stability-Guaranteed Critical Test Curve Algorithm for Generalized Parametric Constraints.” IEEE/ASME Transactions on Mechatronics 23, no. 5 (2018): 2459–2469. doi:10.1109/TMECH.2018.2862144.

[37] Chen, Gang, Yongduan Song, and Yanfeng Guan. “Terminal Sliding Mode-Based Consensus Tracking Control for Networked Uncertain Mechanical Systems on Digraphs.” IEEE Transactions on Neural Networks and Learning Systems 29, no. 3 (2018): 749–756. doi:10.1109/TNNLS.2016.2636323.

[38] Wang, Yaoyao, Jiawang Chen, Kangwu Zhu, Bai Chen, and Hongtao Wu. “Practical Tracking Control of Cable-Driven Robots Using Adaptive Nonsingular Fast Terminal Sliding Mode.” IEEE Access 6 (2018): 68057–68069. doi:10.1109/ACCESS.2018.2879903.

[39] Fethalla, Nuradeen, Maarouf Saad, Hannah Michalska, and Jawhar Ghommam. “Robust Observer-Based Dynamic Sliding Mode Controller for a Quadrotor UAV.” IEEE Access 6 (2018): 45846–45859. doi:10.1109/ACCESS.2018.2866208.

[40] Kern, John, Claudio Urrea, Rodrigo Mendez, and Guillermo Gonzalez. “Development of an Embedded Control System by Means of DsPIC Applied in a 4 DOF Robot.” IEEE Latin America Transactions 14, no. 5 (May 2016): 2099–2106. doi:10.1109/tla.2016.7530401.

[41] Rosales, C, S Tosetti, C Soria, and F Rossomando. “Neural Adaptive PID Control of a Quadrotor Using EFK.” IEEE Latin America Transactions 16, no. 11 (November 2018): 2722–2730. doi:10.1109/tla.2018.8795113.

[42] Khalifa, Mustafa, A.L.Hussein Amhedb, and Mohammed Al Sharqawi. “Position Control of Real Time DC Motor Using LabVIEW.” Journal of Robotics and Control (JRC) 2, no. 5 (2021). doi:10.18196/jrc.25104.

[43] Okelola, Muniru Olajide, David Oluwagbemiga Aborisade, and Philip Adesola Adewuyi. “Performance and Configuration Analysis of Tracking Time Anti-Windup PID Controllers.” Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika 6, no. 2 (January 2021): 20. doi:10.26555/jiteki.v6i2.18867.