

Development of Omni-Wheeled Mobile Robot Based-on Inverse Kinematics and Odometry

Aghus Sofwan

Department of Electrical Engineering
Universitas Diponegoro

Jl. Prof. Soedarto, SH, Tembalang, Semarang. Jawa Tengah, 024
– 7460057, Indonesia.
asofwan@elektro.undip.ac.id

Hadha Afrisal

¹Department of Electrical Engineering, Universitas Diponegoro

²Center for Biomechanics, Biomaterial, Biomechanics, and
Biosignal Processing (CBIOM3S)

Tembalang, Semarang, Jawa Tengah, Indonesia
hadha.afrisal@elektro.undip.ac.id

Hafidz Rizqi Mulyana

Department of Electrical Engineering
Universitas Diponegoro

Jl. Prof. Soedarto, SH, Tembalang, Semarang. Jawa Tengah, 024
– 7460057, Indonesia.
hafidkiki@students.undip.ac.id

Abdul Goni

Department of Electrical Engineering
Universitas Diponegoro

Jl. Prof. Soedarto, SH, Tembalang, Semarang. Jawa Tengah, 024
– 7460057, Indonesia.
abdghon@students.undip.ac.id

Abstract—This paper presents the development of an omni-wheeled mobile robot based on inverse kinematics and odometry for local and indoor navigation purposes, such as for automatic warehousing in industry or healthcare environment. The robot uses four-wheeled diagonal configuration to conform directional angles of $\alpha_1=45^\circ$, $\alpha_2=135^\circ$, $\alpha_3=225^\circ$, and $\alpha_4=315^\circ$. Inverse kinematics is utilized to drive the robot to a point with specific trajectory and heading angle. Internal wheeled-encoders mounted in each DC-motors are used to read the angular speed and position. This research utilizes odometry technique to estimate the robot's position relative to the initial position. In order to develop a more precise odometry result, we combine the use of wheeled-encoders and an IMU. In order to maintain robot's position relative to the desired position, a PID control is applied to the algorithm. The result of the tests show that the developed omni-wheeled mobile robot is capable of performing locomotion to the desired position and to follow a controlled trajectory by maintaining a minimum error relative to the referenced trajectory.

Keywords—mobile robot, omni wheel, inverse kinematics, odometry, PID control

I. INTRODUCTION

Recent advances in wheeled mobile robot developments have opened many possibilities of application for industrial and medical settings. Mobile robot can be potentially utilized for automatic warehousing either in industrial or healthcare environments, such as for loading, transporting, and unloading fragile items from one place to another place continuously and autonomously [1]. One successful implementation of this robot application has been demonstrated by Amazon's robot, Kiva, which is utilized as an autonomous transport machine for sorting and delivering shipped items [2]. It can travel to a pre-defined navigation route in a planar coordinate and is able to move materials from stocking shelves to the fulfillment zones.

One of the most common system utilized in a wheeled mobile robot is the differential drive system [3]. It is a non-holonomic system which is able to move forward, backward, and turn. However, it has a limitation to change its heading condition and to move to any directions. An efficient mobile robot deployed for a warehousing scenario should be able to reach to any directions without turning or changing its heading angle, therefore a holonomic mobile robot system is more suitable to be used in this setting [4]. By applying an inverse kinematics equation into a holonomic system, a robot can be precisely controlled and validated regards to its trajectory

points [3]. The angular speed and direction from its four wheels produce a total force vector to drive the mobile robots to the desired direction without necessarily changing its heading angle. Hence, this holonomic system of a mobile robot will support a higher mobility than the non-holonomic system [4].

II. MOBILE ROBOT MOVEMENT SYSTEM

A. Inverse Kinematics

Inverse kinematics is the usage of the kinematics equations to determine the joint parameters that provide a desired position for each of the robot's end-effectors. Specification of the movement of a robot so that its end-effectors achieve the desired tasks is known as motion planning. On mobile robot, kinematics equation defined some parts from the robot movement which is velocity of V_x , V_y , and ω into its angular speed for each wheel [6]. Fig. 1 and is the sketch design of an omni-wheeled robot with an X-shaped frame consisting of four wheel so that the angle that is formed on each wheel are $\alpha_1=45^\circ$, $\alpha_2=135^\circ$, $\alpha_3=225^\circ$, $\alpha_4=315^\circ$ on the X axis.

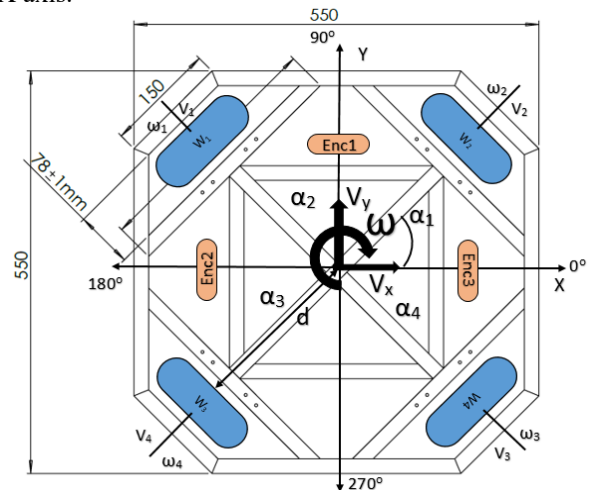


Fig. 1 Wheel and movement vector configuration

Parameters configuration of the velocity are defined in the Table 1. From the parameters values and the formed angle between each wheel and the X axis, each wheel has an additional velocity component towards the X axis and the Y axis in a consequence of the rotational velocity of ω .

Parameters configuration of the velocity are defined in the Table I.

TABLE I.
PARAMETERS CONFIGURATION OF THE VELOCITY

No	Symbol	Description
1	V_x (m/s)	Mobile robot's translational velocity on the x axis
2	V_y (m/s)	Mobile robot's translational velocity on the y axis
3	ω (rad/s)	Mobile robot's angular velocity
4	α_i (degree)	The angle between the omni wheel and the x axis
5	W_i	Symbol for wheels on each side
6	V_i (m/s)	Translational velocity on each wheel
7	ω_i (rad/s)	Angular speed for each wheel
8	d (meter)	Distance between wheel and the center point of the mobile robot
9	enc_i	Symbol for the external rotary encoders

From Table I, The translational velocity on each wheels can be written in equation (1) and (2).

$$V_i = \sin(\alpha_i) V_x + \cos(\alpha_i) V_y + R\theta \quad (1)$$

$$V_i = \omega_i R \quad (2)$$

From equation (1), R is the radius of the omni wheel and θ is the angle that is formed when the robot orientation or the robot heading is changed. Next, by using equation (1) and calculating the inverse kinematics on equation (2), the angular speed for each motors can be derived into equation (3).

$$\omega_i = \frac{1}{R} (\sin(\alpha_i) V_x + \cos(\alpha_i) V_y + R\theta) \quad (3)$$

From equation (3), the inverse kinematics equation for each wheel can be presented on equation (4).

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} \sin(\alpha_1) & \cos(\alpha_1) & R \\ \sin(\alpha_2) & \cos(\alpha_2) & R \\ \sin(\alpha_3) & \cos(\alpha_3) & R \\ \sin(\alpha_4) & \cos(\alpha_4) & R \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \theta \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} \sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) & R \\ \sin\left(\frac{3\pi}{4}\right) & \cos\left(\frac{3\pi}{4}\right) & R \\ \sin\left(\frac{5\pi}{4}\right) & \cos\left(\frac{5\pi}{4}\right) & R \\ \sin\left(\frac{7\pi}{4}\right) & \cos\left(\frac{7\pi}{4}\right) & R \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \theta \end{bmatrix} \quad (5)$$

B. Orientation Angle

An orientation angle is an angle between the initial reference angle and the actual angle. An orientation angle can be described as an euler angle, a three angles introduced by Leonhard Euler to describe the orientation of a rigid body with respect to a fixed coordinate system [8]. They can also represent the orientation of a mobile frame of reference in physics or the orientation of a general basis in 3-dimensional linear algebra that consists of roll, pitch, and yaw and can be written as ϕ , θ , and ψ . The composition of each axis which is x, y, and z on the roll, pitch, and yaw axis can be represented on a rotational matrix [8].

The reference frame for the orientation angle is the earth reference frame in a north-south frame for the x axis, east-

west frame for the y axis, and up-down frame for the z axis. On this mobile robot design, a 9 degrees of freedom IMU that consists of a 3-axis gyroscope, 3-axis accelerometer, and 3-axis magnetometer is used to acquire only the yaw orientation angle. This is because the robot is moved in a flat- two-dimensional surface only. Nevertheless, the calculation process to receive the yaw angle is a cascade process, so it needs the calculation from every sensors. The yaw angle is obtained from the gyroscope and magnetometer, but first it has to collect the data from the accelerometer.

The angle readings using an accelerometer is shown in equation (6).

$$\begin{bmatrix} fx \\ fy \\ fz \end{bmatrix} = g \begin{bmatrix} \sin \theta \\ -\cos \theta \sin \phi \\ -\cos \theta \cos \phi \end{bmatrix} \quad (6)$$

A 3-axis accelerometer produces an output of fx , fy , and fz as a gravity force on x, y, and z axis respectively. Pitch and roll orientations can be acquired by deriving equation (6) into equation (7) and (8).

$$\phi = \tan^{-1} \left(\frac{fy}{fz} \right) \quad (7)$$

$$\theta = \tan^{-1} \left(\frac{-fx}{\sqrt{fy^2 + fz^2}} \right) \quad (8)$$

The angle readings using a gyroscope is shown in equation (9) with p, q, and r are representing the body reference axis.

$$\begin{bmatrix} \phi \\ \theta \\ \varphi \end{bmatrix} = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (9)$$

The yaw angle reading on a 3-axis magnetometer is needed to obtain the earth's magnetic field data on the robot as (m_x , m_y , and m_z). Relations between the magnetometer readings and the earth's magnetic field (m_x , m_y , and m_z) that consists of north, east, and down component can be written on equation (10).

$$\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \phi & \cos \theta \sin \phi & -\sin \theta \\ \sin \theta \sin \phi \cos \phi - \cos \theta \sin \phi & \sin \theta \sin \theta \sin \phi + \cos \theta \sin \phi & \sin \theta \cos \theta \\ \cos \theta \sin \theta \cos \phi - \sin \theta \sin \phi & \cos \theta \sin \theta \sin \phi + \sin \theta \cos \phi & \cos \theta \cos \theta \end{bmatrix} \begin{bmatrix} m_N \\ m_E \\ m_D \end{bmatrix} \quad (10)$$

The yaw angle reading from the magnetometer sensor can be produced by deriving equation (10) into equation (11).

$$\varphi = \tan^{-1} \left(\frac{-m_y \cos \phi + m_z \sin \phi}{m_x \cos \theta + m_y \sin \theta \sin \phi + m_z \sin \theta \cos \phi} \right) \quad (11)$$

C. Mobile Robot Odometry System

Odometry is the use of data from motion sensors to estimate change in position over time. It is used to estimate the robot's position relative to a starting location [6]. In this

paper, a three external rotary encoder and an IMU are used to get the robot's traveled distance towards X and Y coordinate. The first and the second external rotary encoder are used to calculate the robot's traveled distance towards x and y coordinate, while to calculate the robot's orientation or the robot heading, the second and the third external rotary encoder is used. The data will be combined with the yaw orientation reading from the IMU sensor. The configuration of the external rotary encoder position can be seen on Fig. 1. There are three main parameters that are used in calculating robot's position/ coordinate. The parameters are the encoder wheel diameter (R_{enc}), the total encoder resolution ($Resolution_{enc}$), and the pulse that is generated by the encoder (pulse). The circumferences of the encoder wheel (K_{enc}) can be calculated by multiplying the encoder wheel diameter (D_{enc}) with π (phi) as seen on equation (12):

$$K_{enc} = R_{enc} \times \pi \quad (12)$$

From equation (12), the encoder traveled distance on the Cartesian coordinate can be calculated using equation (13) and (14).

$$X_{distance} (cm) = \frac{pulse_x}{resolution_{enc1}} \times K_{encoder1} \quad (13)$$

$$Y_{distance} (cm) = \frac{pulse_y}{resolution_{enc2}} \times K_{encoder2} \quad (14)$$

$$\vartheta (radian) = \frac{(Enc2_{cm} + Enc3_{cm})}{2 \times d_{enc}} \quad (15)$$

Equation (15) is used to obtain the heading alteration of the robot when it moves to the desired position or coordinate. Equation (15) is obtained by taking the mean of the second and the third external rotary encoder. The calculation process to produce ($Enci_{cm}$) is same as equation (13) and (14). Furthermore, the result from equation (15) is multiplied by the distance between the external rotary encoder and the center point of the robot (d_{enc}) as seen on equation (16).

$$\vartheta (degree) = \vartheta (radian) \times 57.298 \quad (16)$$

The robot's heading value from the external rotary encoder sensor will be fused with the robot's heading value from the IMU which is the yaw value using a complementary filter. The yaw readings using external rotary encoder and the IMU both have same characteristics, that's why the gain constants are same for both sensor readings which is 0.5 as seen on equation (17).

$$\psi = 0.5 (\vartheta_{rotary}) + 0.5 (\varphi_{IMU}) \quad (17)$$

To get the actual robot traveled distance can be calculated using equation (18) and (19).

$$X_{actual} = X_{distance} \times \cos(\psi) + Y_{distance} \times \sin(\psi) \quad (18)$$

$$Y_{actual} = Y_{distance} \times \cos(\psi) - X_{distance} \times \sin(\psi) \quad (19)$$

III. DESIGN SYSTEM OF MOBILE ROBOT

A. Mechanical Design

The mobile robot frame is designed as an octagonal-shaped with the outer size of 550 mm x 550mm. The mobile robot frame is made from a 2x2 cm stainless steel (hollow, not solid). In its making, to make a strong and precision frame it needs the welding technique process to connect each parts

of the stainless steel. The mechanical design is shown in Fig. 2.

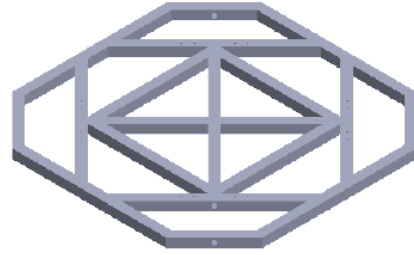


Fig. 2 A 3D mechanical design

B. Electrical Design

The electronic design for this mobile robot is using three microcontroller. The first microcontroller act as a master and the other two as a slave. The mobile robot is using four internal rotary encoder sensors, three external rotary encoder sensors, and an IMU. Four internal rotary encoder are connected to the second slave microcontroller, three external rotary encoder are connected to the master, and then an IMU is connected to the first slave microcontroller. The electronic design can be seen in Fig. 3.

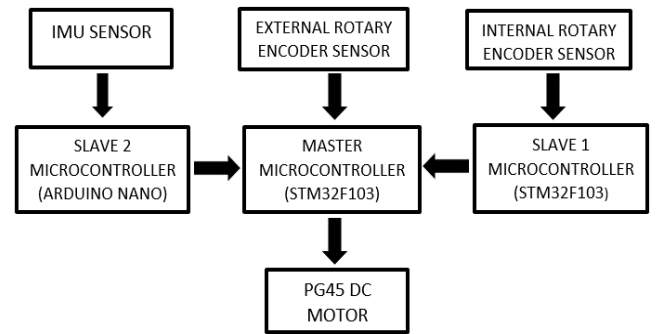


Fig. 3 Electronic block diagram

The arrows on the Fig.3 are represented the communication process and the readings from sensors on each microcontroller. The first slave microcontroller is only used to process the yaw value from the IMU readings. The second slave microcontroller is used to process each DC PG45 motor's angular speed readings. The results from processes on each microcontroller will be send to the master microcontroller using a serial communication. The master microcontroller is also used to processes the odometry from the external rotary encoder, combining the data from each slave microcontroller, as well as controlling the DC PG45 motor, performing the inverse kinematics process and also the PID control.

C. PID Control

A smart robot certainly has the ability to maintain its set point. In designing this mobile robot, a PID control method is used to control the mobile robot. The feedback signals for the PID control are coming from each sensors readings. On this research, three external rotary encoder, four internal rotary encoder, and an IMU are used. The controlling process on this research is divided into two sub-system, which are the DC motor speed control and the robot's actual positional control.

The control signal for the DC motor speed control is generated. The PID block diagram is shown in Fig. 4.

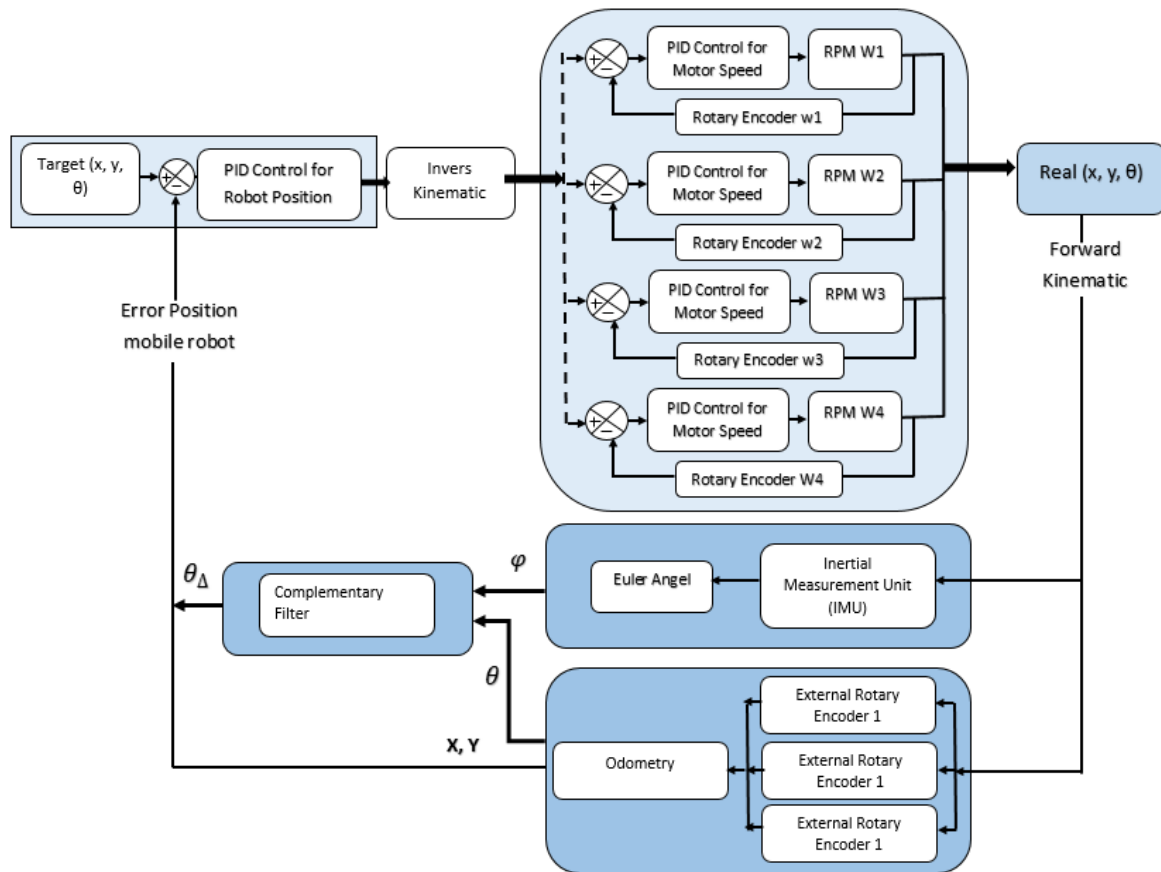


Fig. 4 The PID block diagram

From the difference between the set point value and the internal rotary encoder sensor readings. The control signal for the robot's actual positional control is generated from the external rotary encoder and the IMU readings. The output from the robot positional control are the coordinate and the heading of the robot.

In designing the mobile robot, a PID control for each motors are separately placed. A PID for DC motor 1, 2, 3 and 4, yet the parameters value for proportional control (K_p), integral (T_i), and derivative (T_d) are same because of the similarity on the DC motors characteristics. The parameters values determination are using a Ziegler Nichols method, while the parameters value for the robot positional control are generated from the trial error results.

IV. TESTING AND IMPLEMENTATION

A. DC Motor Speed Control Test

PID control is needed to fix the system response. To get a good system response, it needs a tuning for the PID parameters value which are K_p , T_i , and T_d . On this paper, the Ziegler-Nichols 2 tuning method is chosen in determining the PID parameters due to the ease and the simplicity on the calibration. The test is done using only one DC motor as a sample with a 100 rpm set point. The first step on tuning the Ziegler-Nichols 2 method is by adding the K_p value until the

system is oscillating continuously, after that set the value as the K_u value. The next step is determining the P_u value, which is an oscillating period when the K_u reach its maximum value. After acquiring the K_u and P_u value, the next step is multiplying both values with the Ziegler-Nichols 2 method constant. The results from the tuning process are $K_p= 5$, $T_i= 2.8$, and $T_d= 0.6$. The response from the implementation on the DC motor PID control can be seen in Fig. 5.

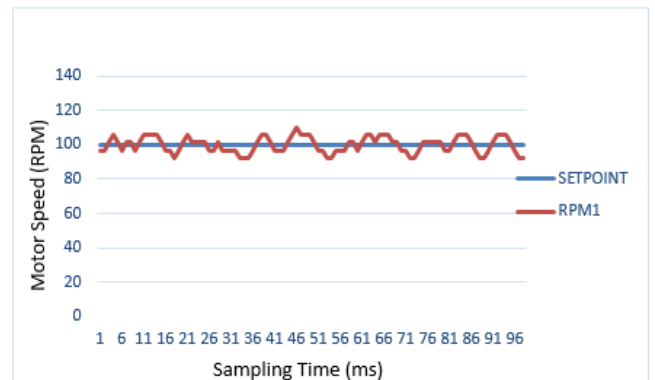


Fig. 5 The result from the DC motor PID control

The PID control is also used in the other three DC motors. The test for each motors are done by given each motors a speed set point of 110 rpm until 200 rpm. The variation of the speed set point is aiming to observe the system response whether it can follow the set point or not. Next, a hundred data from each internal rotary encoder readings is taken and then average it from each set points and motors. The data that are presented on Table. II are acquired from the average of a hundred data from each motors and then mean it from 10x testing.

TABLE II.
Average motors speed towards the rpm set points

Setpoint (RPM)	Average Of Motor Speed			
	Motor 1 (RPM)	Motor 2 (RPM)	Motor 3 (RPM)	Motor 4 (RPM)
110	110.1	110.0	110.1	110.0
120	120.0	119.8	119.9	120.0
130	129.6	129.9	129.9	129.7
140	140.1	140.1	140.2	140.3
150	149.9	149.5	149.6	149.3
160	160.0	160.1	160.0	159.8
170	169.9	169.9	169.9	169.9
180	180.3	180.4	180.2	180.3
190	190.2	189.7	189.8	189.7
200	199.8	200.3	200.0	200.1

B. Mobile Robot Positioning Control Test

This part is a whole system testing using the control algorithm that is gained from Fig. 4. The test is done on the custom arena made from a 2400x2400 x 9 mm wood layer. The arena is following the standard for the ABU Robocon arena. The test is performed in a three different methods, which are centered turing, driving forward, and a trajectory.

1. Centered Turing Test

There are two steps in reaching the final position using the centered turing method. First, the robot will adjust its orientation following the calculation of the resultant using Pythagoras equations from the x and y desired coordinate. In this test, the desired coordinate is (100,100) cm. The result from the test is exhibited in Fig. 6.

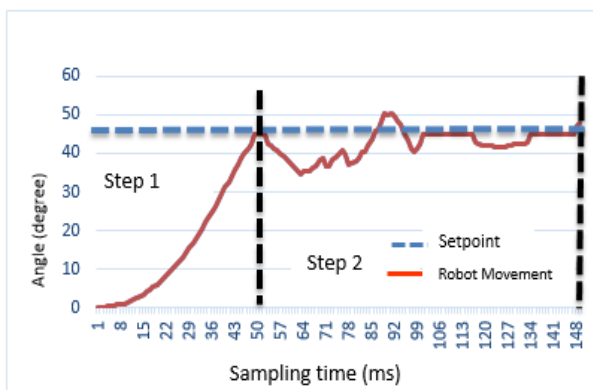


Fig.6 The result from the centered turing method

Fig. 6 shows the first step which is the mobile robot adjusting its heading by rotating 45° following the resultant calculation of the x and y coordinate input. On the second step, the robot maintains its heading on 45° when the robot moves forward to the final destination. In this test, there is an error of 4° headings from the desired 45° and an error of 1 cm on its final destination.

2. Driving Forward Test

The different between the driving forward and the centered turing test is the mobile robot will not calculate its heading or its distance resultant on the driving forward test. The robot will move to the desired position without changing its heading or yaw value. In driving forward test, a (125, 100) set point coordinate is given. The first step is the mobile robot will move forward as far as 125 cm on the x axis and then continued by moving laterally as far as 100 cm on the y axis. The test results using driving forward method can be seen in Fig. 7.

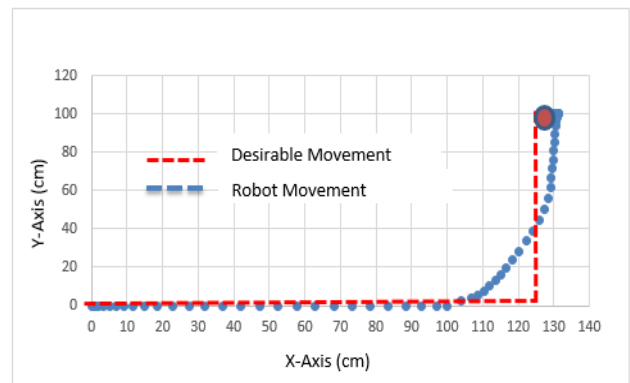


Fig.7 The result from the driving forward method

Fig. 7 shows that the mobile robot is moving starts from the x axis and then the robot will move on the y axis. When the robot moves on the x axis, it can be seen that the robot already moves towards y axis before it reaches the desired x coordinate. The robot starts moving laterally from x= 100cm instead of x= 125cm so that's why there is a 5 cm offset from the set point on its x coordinate final position, while when the robot moves on the y axis, its final position is y= 102 cm instead of y= 100 cm. This state is happens because of some factors, which are there is a slip on the external rotary encoder wheel or it needs another tuning on the PID control parameters.

3. Trajectory Test

The trajectory test is done by giving the robot some trajectories that needs to be followed by the mobile robot. The movement process can be determined by using an inverse kinematics equations on equations (5). By setting the speed vectors value (V_x , V_y , and ω) or the (x, y) desired coordinates, the angular speed and the directions from each motors can be produced. A Trajectory is designed by given the desired coordinate value on the Cartesian axis which is x and y.

a. Square Trajectory

An array contains of coordinates are given to produce the square trajectory. Started from the first array, with the initial

position $(x, y) = (0, 0)$ and then followed by the next array. The Coordinate for the square trajectory is $X = \{0, 120, 120, 0, 0\}$ and $Y = \{0, 0, -120, -120, 0\}$. The result from the square trajectory can be seen in Fig. 8.

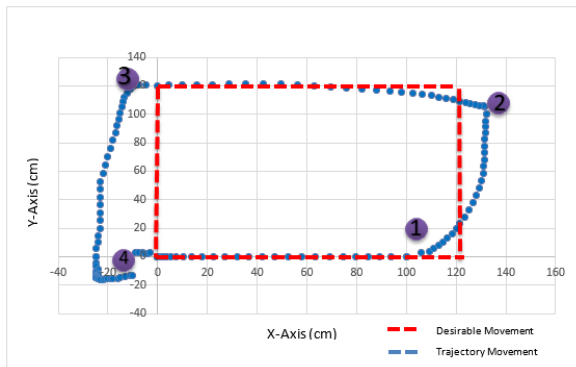


Fig.8 The result from the square trajectory

Fig. 8 shows the actual trajectory that has been traveled by the mobile robot. On the first step, the robot will move from the $(0, 0)$ coordinate to the $(120, 0)$ coordinate. The robot moved quite straightaway on the first three-quarter but when it approaching the set point on the $(108, 3)$ coordinate, the robot start to move laterally on the y axis before it reached its set point on $(120, 0)$ coordinate as seen on the first dot. On the second step, the robot moves towards the $(120, 120)$ and deviated far enough from the desired trajectory and then move towards the x axis before reaching its set point as seen on the second dot. On the third step, the robot is able to move to its desired trajectory when heading to the $(0, 120)$ coordinate, however when it start to move to the fourth step, the robot having an offset as seen on the third dot. When the robot moves in the fourth step, there is an offset again as seen on the fourth dot and the robot stop moving on the final coordinate of $(-11.27, -10.3)$.

a. Triangle Trajectory

An array contains of coordinates are given to produce the triangle trajectory. Started from the first array, with the initial position $(x, y) = (0, 0)$ and then followed by the next array. The Coordinate for the triangle trajectory is $X = \{0, 120, 60, 0\}$ and $Y = \{0, 0, 60, 0\}$. The result from the triangle trajectory can be seen on Fig. 9.

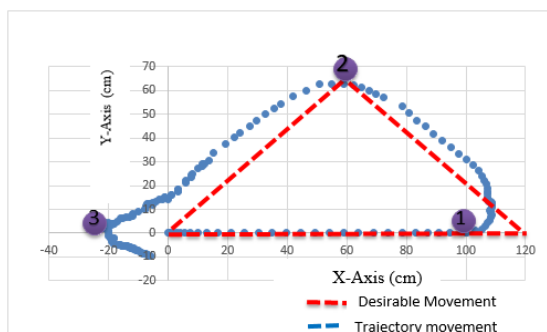


Fig.9 The result from the square trajectory

Fig. 9 shows the actual trajectory that has been traveled by the mobile robot. On the first step, the robot will move from the $(0, 0)$ coordinate to the $(120, 0)$ coordinate. The robot moves quite straightaway but when it approaches the

set point on the $(107.4; 3.1)$ coordinate, the robot starts to move to the second step on $(60, 60)$ coordinate as seen on the first dot. On the second step, the robot moves towards the $(60, 60)$ coordinate and it can maintain its desired position by producing the final $(61.2; 63.2)$ coordinate, as seen on the second dot. On the third step, the robot having a deviation from the desired trajectory with an offset on the $(-22.26; 6.2)$ as seen on the third dot. The robot is able to correct its deviation by moving to the $(-5.6; -10.1)$ coordinate before stops.

V. CONCLUSION

From the test, it can be concluded that:

1. PID control parameters tuning using the Ziegler- Nichols 2 and the trial and error method can produce a steady response for each motors with $K_p = 5$, $T_i = 2.8$, and $T_d = 0.6$.
2. The centered tuning test produces a 4° errors on the final heading set point and a 4.2 cm offset on the final traveled distance set point.
3. The driving forward test produces a 5cm errors on the x coordinate set point and a 2 cm errors on the y coordinate set point.
4. The square trajectory test produce a maximum absolute error of 10.4 cm on the x axis and 12.24 cm on the y axis.
5. The triangle trajectory test produce a maximum absolute error of 22.1 cm on the x axis and 12.3 cm on the y axis.

REFERENCES

- [1] Bogue, R, "Growth in e-commerce boosts innovation in the warehouse robot market", *Industrial Robot*, Vol. 43 No. 6, pp. 583-587, 2016-0194.
- [2] Jun-tao Li, Hong-jian Liu. "Design Optimization of Amazon Robotics, Automation, Control and Intelligent Systems", Vol. 4, No. 2, pp. 48-52, 2016.
- [3] S. Chamberland, É. Beaudry, L. Clavien, F. Kabanza, F. Michaud, dan M. Lauriay, "Motion Planning for an Omnidirectional Robot with Steering Constraints," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.
- [4] Taheri, Hamid, Bing Qiao, and Nurallah Ghaeminezhad, "Kinematic Model of a Four Mecanum Wheeled Mobile Robot", *International Journal of Computer Applications (0975 – 8887)*, Volume 113 – No. 3, March 2015.
- [5] Tatar, Mihai Olimpiu., "Design and Development of an Autonomous Omni-Directional Mobile Robot with Mecanum Wheels," 2014. pp. 978-1-4799-3732-5.
- [6] Zhang, Yunan, Wang Shuangshung, Zhang Jian, "Research on Motion Characteristic of Omnidirectional Robot based on Mecanum Wheel", *2010 International Conference on Digital Manufacturing & Automation Research*, 2010. pp. 978-0-7695-4286-7/10.
- [7] Park, J, Kim, S. 2010. "Driving Control of Mobile Robot with Mecanum Wheel using Fuzzy Inference System", *World Academy of Science Engineering Technology*, Vol.6, pp.2519-2523, 2010.
- [8] Malyavej, V. Kumkeaw, W.; Aorpimai, M., "Indoor robot localization by RSS/IMU sensor fusion," *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTICON), 2013 10th International Conference*, pp.1-6, 15-17 May 2013.
- [9] M. Jaradat and M. Abdel-Hafez, "Enhanced, Delay Dependent, Intelligent Fusion for INS/GPS Navigation System", *IEEE Sensors J.*, vol. 14, no. 5, pp. 1545-1554, 2014.
- [10] Taehee Lee, Joongyou Shin, Cho, D, "Position estimation for mobile robot using in-plane 3-axis IMU and active beacon," *in Industrial Electronics, IEEE International Symposium*, vol., no., pp.1956-1961, 5-8 July 2009.