Positioning and Maneuver of an Omnidirectional Robot Soccer

Khasia Vidra Utama
Department of Electrical Engineering
Batam State Polytechnic
Batam, Indonesia
khasiavidrap@gmail.com

Senanjung Prayoga
Department of Electrical Engineering
Batam State Polytechnic
Batam, Indonesia
senanjung@polibatam.ac.id

Rizky Pratama Hudhajanto
Department of Electrical Engineering
Batam State Polytechnic
Batam, Indonesia
rizkypratama@polibatam.ac.id

Rifqi Amalya Fatekha
Department of Electrical Engineering
Batam State Polytechnic
Batam, Indonesia
rifqi@polibatam.ac.id

Daniel S Pamungkas
Department of Electrical Engineering
Batam State Polytechnic
Batam, Indonesia
daniel@polibatam.ac.id

***Abstract*—To control omnidirectional robot is a challenging problem. This paper introduces the implementation and the method to control the robot soccer which uses omnidirectional wheels. Forward, inverse kinematics algorithms and PID controller have also used an arm controller, encoder and compass sensors are utilized. This paper shows that the proposed system has the potential to control the position and maneuver of the robot.**

Keywords—inverse kinematics, forward kinematics, PID, mobile robot, omnidirectional.

I. INTRODUCTION

Indonesian wheeled soccer robot competition that has been held in 2017 is referred to Robo Cup Middle Size League (MSL). The robot will play like a football played by a human. In this competition, most of the robots are used Omni wheel robots.

An omniwheel robot is a mobile robot which uses wheels which can move in any direction and any orientation [1,2]. However, omniwheel robot type has problems especially to control and to model this type of mobile robot. [3].

To control the omniwheel robot, several researchers use the dynamic model. [4]. Others researchers utilize the kinematic models [5]. However, dynamic and kinematic methods assume that no slippage occurs on the robot on those methods.

To compensate for the slippage, Jeong-Hyeong Lee and Seul Jung use kinematics derivatives in the global frame [6]. They use a compass and encoder sensor to locate the robot. The function of the compass sensor is to reduce the error which occurs from the calculation of the encoder. One of the reasons the error occurs because the slip happens. Moreover, they used a PID controller to control the movement of the robot. While other research; Rifky Afriza, using the inverse equation combined with the feedback forward kinematic PID-based [7] mentioned that for checking robot position is using the Forward Kinematics equation.

This paper proposes a position and maneuver control method for the omnidirectional robot system with a combination between forwarding, inverse kinematics, and PID controller. Also, this system employs the compass and encoder sensors.

This paper is organized as follows. In section 2, the proposed method is described. The experimental results and discussion are explained in section 3. Finally, the conclusion is presented in section 4.

II. METHOD

2.1 Hardware

Figure 1 shows the devices of the proposed system. Three rotary encoders are employed to count the rotation of the wheels. The outputs of these encoders are transferred to the controller. This controller processed those input and send to the PC. This process is to generate the path of the robot. The controller also transmits the signal to the motor controller to move the motor.

For the experiments, a robot using six Omni wheels is employed. The wheels in the robot consist of two types of dimensions. There are three Omni wheels with a diameter of 10 cm. Other three wheels have diameter 5 cm. The bigger wheels are placed on the outside an equilateral triangle base of the robot. Each wheel has distance 19 cm from the robot center to the wheel center. Then the width of the angle formed between each wheel is 120 degrees.

The smaller wheels are placed inside the base of the robot. These wheels are also equilateral. The center robot's distance to each wheel center is 8 cm. The width of the angle formed between each wheel is 120 degrees as well. The design of the robot can be seen in Figure 2.

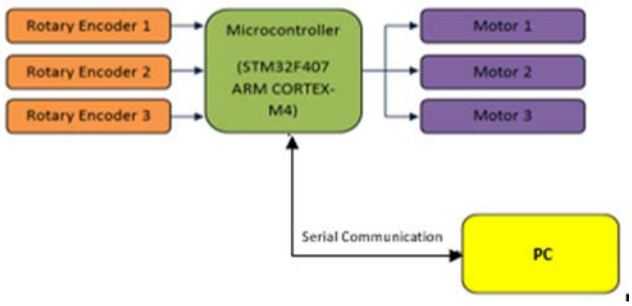


Fig. 1. Block diagram of a robot

.

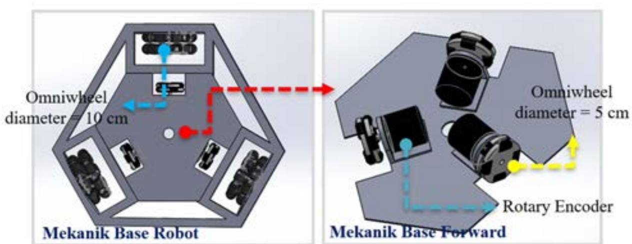


Fig. 2. Mechanical design of the robot

STM32F407 is a microcontroller based on ARM Cortex-M4 is used for the controller of this system. This controller has a processing time speed of up to 168 Mhz. Moreover, this device has additional features such as an accelerometer, audio sensor, omnidirectional digital microphone, audio DAC with the integrated class D speaker driver, also has two pushbuttons, one of which functions as a reset button and user key (can be set as input). [8].

Rotary encoders and compass sensor are utilized in this system. To count the rotation of the wheels, the encoders are used. The encoders are coupled with 5 cm diameter wheels as seen in Figure 1. The MPU 6050 is used for the compass sensor. This sensor is known as Micro electro mechanical systems (MEMS). This sensor is an accelerometer and a gyro in a chip.

2.2 Software

This system proposed a combination between inverse kinematics and forward kinematics. PID control is used to control the movement of the robot to appropriate with the track that we define.

Inverse kinematics is a matrix formula that determines the linear velocity. This algorithm required an angular velocity of each wheel. The system designed with a configuration such as a Figure 3. The Omniwheels positions are in symmetrically. With the center of the robot.

The $\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3$ determines the speed of each wheel where Position $\dot{\phi}_1$ (1/6 π) = 30°, $\dot{\phi}_2$ (5/6 π) = 150°, $\dot{\phi}_3$ (15/10 π) = 270°. The matrix formulas used in kinematic inverse are shown in (1) [2].

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_1 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta + a_1) & \cos(\theta + a_1) & R \\ -\sin(\theta + a_2) & \cos(\theta + a_2) & R \\ -\sin(\theta + a_3) & \cos(\theta + a_3) & R \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & 0 \\ 0 & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_L \\ \dot{y}_L \\ \dot{\theta} \end{bmatrix} \quad (1)$$

Where :

$\dot{\phi}$ = Wheel speed (radians per second)
$\dot{x}_L$= Speed on the x axis (cm per second)
$\dot{y}_L$= Speed on y axithe s (cm per second)
$\dot{\theta}$ = Robot angle velocity (radians per second)
R = wheel distance to the robot mass center (cm)
r = wheel radius (cm)

Forward kinematic is a matrix formula that determines the position and direction of the robot as it moves. It is designed with a configuration such as a Figure 4. The figure shows the location of mounting three symmetrical wheels. The omniwheels position is symmetrical to the center of the robot attached, located at the midpoint of the robot. Position $\dot{\phi}_1$ (1/6 π) = 180°, $\dot{\phi}_2$ (5/6 π) = 300°, $\dot{\phi}_3$ (15/10 π) = 60°

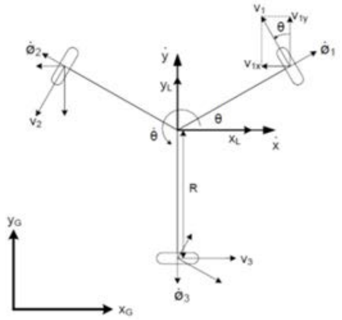


Fig. 3. Diagram Inverse Kinematic

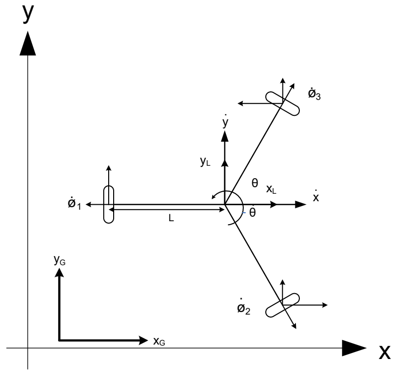


Fig. 4. Diagram forward kinematic

To obtain the desired movement can be calculated using the Jacobian matrix equation, see (2).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = R \begin{bmatrix} -\frac{2\,\mathrm{Sin}[\psi]}{3} & -\frac{\mathrm{Cos}[\psi]}{\sqrt{3}} + \frac{\mathrm{Sin}[\psi]}{3} & \frac{\mathrm{Cos}[\psi]}{\sqrt{3}} + \frac{\mathrm{Sin}[\psi]}{3} \\ -\frac{2\,\mathrm{Cos}[\psi]}{3} & \frac{\mathrm{Cos}[\psi]}{3} + \frac{\mathrm{Sin}[\psi]}{\sqrt{3}} & \frac{\mathrm{Cos}[\psi]}{3} - \frac{\mathrm{Sin}[\psi]}{\sqrt{3}} \\ \frac{I}{3\,L} & \frac{I}{3\,L} & \frac{I}{3\,L} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (2)$$

Where :

$\dot{x}$ = Speed on the x axis (cm per second)
$\dot{y}$ = Speed on y the axis (cm per second)
$\dot{\psi}$ = Robot angle velocity (radians per second)
R = Radius The robot wheel (cm)
L = Distance of robot center to wheel center (cm)
$\dot{\theta}$= Wheel rotation speed (rotation per second) [2].

The input of the kinematic forward is the rotation speed of wheel 1 ($\dot{\theta}_1$), wheel 2 ($\dot{\theta}_2$), and wheel ($\dot{\theta}_3$). The result of output is velocity on the x axis ($\dot{x}$), y ($\dot{y}$) per second and angular velocity ($\dot{\varphi}$) per second. Then the result is converted by integrating with time as seen in equation (3). This process is to achieve the position of robot movement in the direction form.

$$\begin{aligned} x &= x + \int \dot{x}\, dt \\ y &= y + \int \dot{y}\, dt \\ \varphi &= \varphi + \int \dot{\varphi}\, dt \end{aligned} \quad (3)$$

Figure 5 is a block diagram of how the system is performing. The destination position defines as the x-axis, the y-axis, and the robot angle (ψ). These are the setpoints (Esp). While the process variable (Ym) is the x,y and ψ actual. The values of x and y obtained from the encoder sensors which processes with from the forward kinematics, while the ψ actual is the information from the compass sensor. Those values Esp and Ym then processed in PID controller. Afterward, the signal from PID become inputs for the inverse kinematic equation. To be converted to the speed of each motor in the robot for the robot to move. We will examine the proposed method in the following section.

III. Result and Discussion

Before the experiments are conducted, the values of the PID controller must be obtained. The Ziegler-Nichols tuning method is used to acquire those values.

In the next experiment, the robot has tasks to follow a square path. The robot must move counterclockwise direction. This path has 2000 cm x 2000 cm dimension. First, the robot must go forward in the y-axis, then the robot moves to the left. After that, the robot goes backward then going to the right and stop in the initial position. The results from the reading of the encoder sensor are shown in Figure 6(a). It shows that the robot is moving in almost square track. However, in the actual position is different. The Figure 6(b) shows the actual location.

The result from this experiment shows that there is an error in each movement. The average error when the robot moves forward in the Y-axis is 2.31%. While the robot moves to the right along the X-axis, the failure is 3.18%. The error 9,17% and 23.9% are the average errors for the next movements. The movement of each axis against time is shown in figure 7.

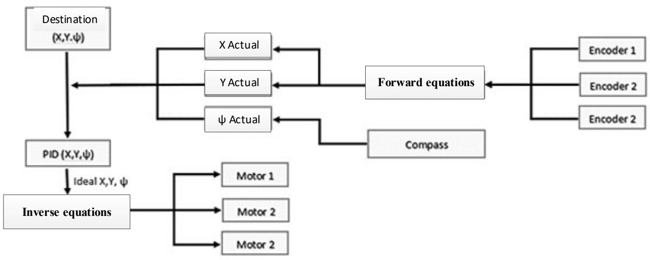
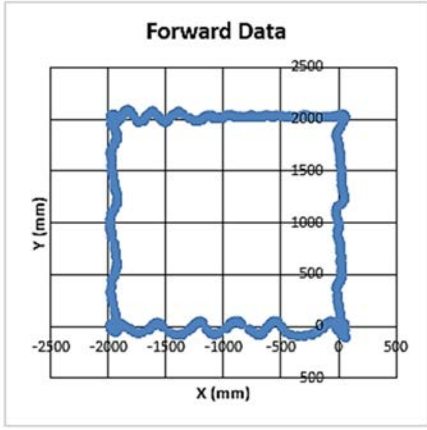
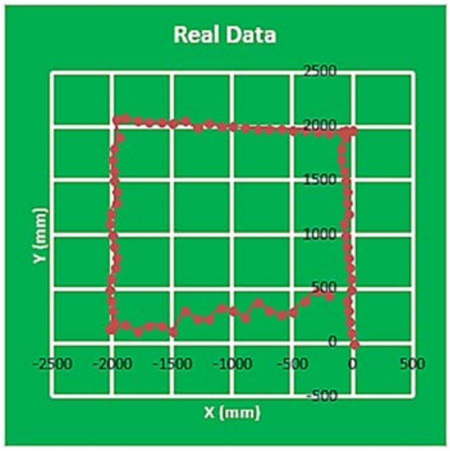


Fig. 5. Block diagram of the algorithm

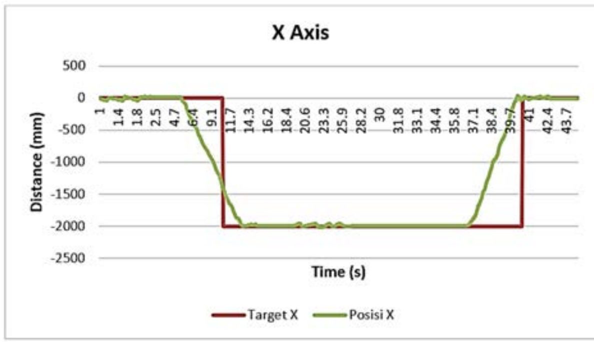


(a)

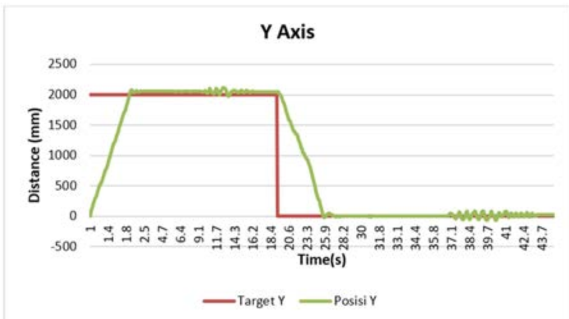


(b)

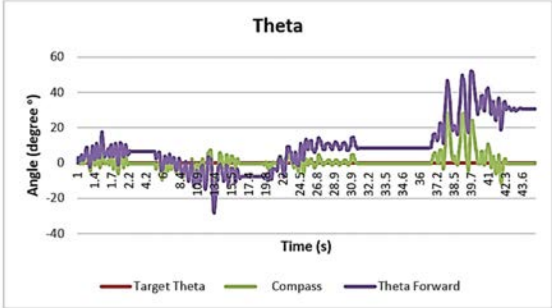
Fig. 6. Robot position results in a. Encoder Reading b. Measurement on the field



a. Robot position on the x-axis



b. Robot position on the y-axis



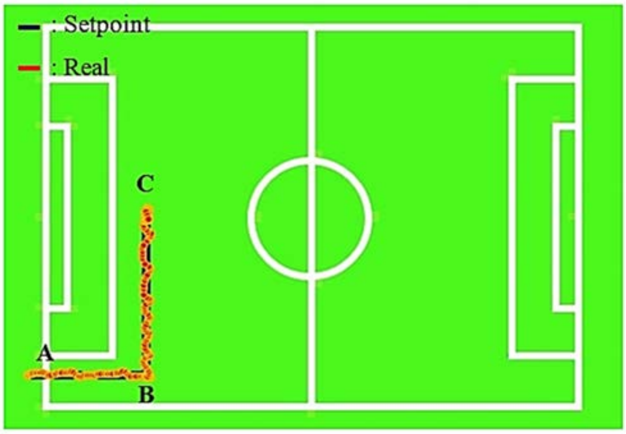
c. Direction of robot

Fig. 7. Graph results for the position and direction of the robot

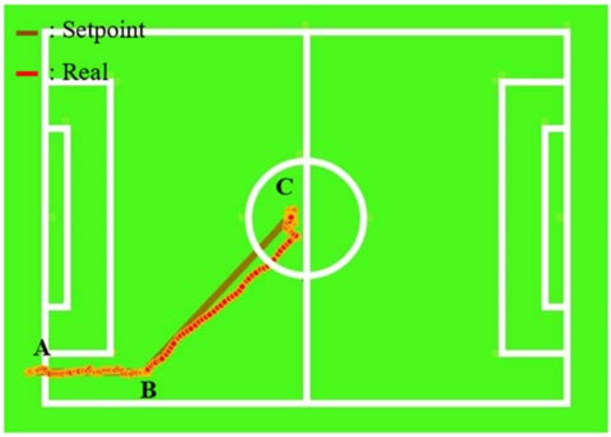
Robot task in the second experiment is to move along the path as seen in Figure 8. This task is to simulate the robot in a defense position. The robot starts moves from Point A (0, 0), to point B (0, 2000), then turn to point C (-2500, 2000) all in cm. The direction of the robot is instructed in 0 degrees all the time. The sensors show that point A is in (0, 0) while B is in (20.17, 2002.989), and C (-2468.24, 2031.077). Whereas, the real measurement obtain that the point A is (0, 0), B (30, 1981), and C (-2301, 2050). While the direction of a robot when the robot reaches the point B is -0.35 and point, C is 7.41.

The next test is to make the robot move as shown in Figure 9. The movement in this attempt is to simulate a robot as a midfielder. Point A ((0, 0) this is the starting position of the robot, next robot will moves to point B (0, 2000), continue to the point C (-2500, 4500). This movement is done in a constant direction (ψ) 0 degree. The results from the sensors detect positions A is at (0, 0), B (-4.145, 2027.047), and C (-2501.79, 4548.606). While the results from the measurement in the field are A (0, 0), B (-83, 1912), and C (-2198, 4612.5). Moreover, the direction of a robot from the compass when it reaches the point B is the same with point C is -0.35.

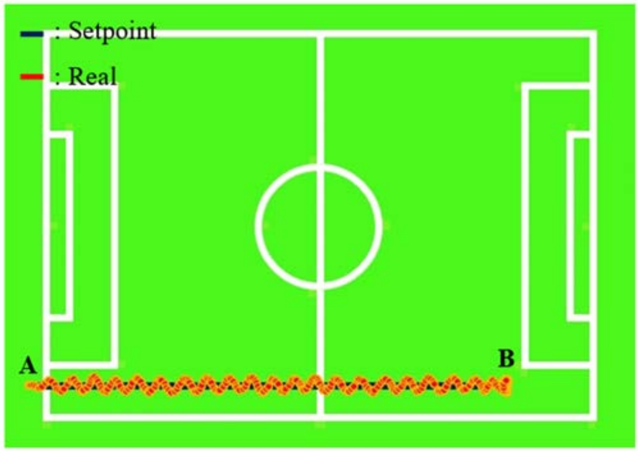
The last task of the robot is to move in a straight path. This trial is to simulate the robot as a striker. Figure 10 shows the task of the robot. The robot begins to move from the starting point, A (0, 0) to the final point, B (0, 7500) The robot direction is 0 degree. The results come from the sensor for A and B are (0, 0), (-85.012, 7862.197) respectively. While when the measurement is held the results are point A is in (0, 0), and B is in (-723, 7923). The direction of the robot at the point B is -15.18.



(a)



(b)



(c)

Fig. 8. Positioning robot as (a) a defender (b) a midfielder (c) a striker

IV. Conclusion

This paper introduces the method to control robot which utilizes omniwheels. Forward, inverse kinematics and PID control are used for the algorithm. The encoder and compass are used in this method. This method applies in the robot soccer. The results show that the error still occurs especially with continuous movement. However, this method is potential to be used for robotic soccer competition.

ACKNOWLEDGMENT

This study was supported by Politeknik Negeri Batam

REFERENCES

[1] F. G. Pin and S. M. Killough, A new family of omnidirectional and holonomic wheeled platforms for mobile robots, IEEE Trans. on Robotics and Automation, 10 (4) (1994) 490-489.

[2] H. C. Huang and C. C. Tsai, FPGA Implementation of an embedded robust adaptive controller for the autonomous omnidirectional mobile platform, IEEE Trans. on Industrial Electronics, 56 (5) (2009) 1604-1616.

[3] VIET, T; et al. Tracking control of a three-wheeled omnidirectional mobile manipulator system with disturbance and friction. Journal of Mechanical Science & Technology. 26, 7, 2197-2211, July 2012. ISSN: 1738494X.

[4] Tsai, C.-C., Huang, H.-C., Wang, T.-S., and Chen, C.-M. (2006). System design, trajectory planning and control of an omnidirectional mobile robot. In 2006 CACS Automatic Control Conference.

[5] Rojas, R. and Forster ¤, A. G. (2006). Holonomic Control of A Robot with an Omni-directional Drive. BottcherIT ¤ Verlag, Bremen.

[6] Jeong-Hyeong Lee and Seul Jung, “Global Position Tracking Control of an Omni-directional Mobile Robot Using Fusion of a Magnetic Compass and Encoders,” intelligent systems and emotional engineering laboratory(ISEE), Mechatronics Engineering Departement, Chungnam National University,Daejeon, 305-764 KOREA.

[7] Rifky Afriza, “Position Control pada Mobile Robot Omnidirectional Wheel Kiwi Drive menggunakan persamaan Inverse dengan Feedback Forward Kinematic berbasis PID” , B.S. Thesis, Politeknik Negeri Batam, 2016.

[8] Jaouadi, Aymen et al. “Adaptive STM32F4 Microcontrollers - Case of Flexible Smart Meters.” *SMARTGREENS* (2014).