Trajectory Tracking Control of Three Wheeled Omnidirectional Mobile Robot Using Proportional Controller

Pipit Anggraeni[1], Nur Jamiludin Ramadhan[2], Miftahul Huda[3]

Department of manufacturing automation
and mechatronics engineering
Bandung Polytechnic for Manufacturing
Bandung, Indonesia
`pipit anggraeni@polman-bandung.ac.id, nj ramadhan@polman-bandung.ac.idng, miftahulhudaa42@gmail.com`

Abstract. Polebot is the educational robot developed by Bandung Polytechnic of Manufacturing which has a type of omnidirectional mobile robot with three Omni wheels assembled at an angle of 120 degrees. Stability of movement is one of the important things in robots. This study aims to make Polebot move according to the input trajectory that has been created. By entering the coordinates on the x and y axes that make up a trajectory, it can be seen what percentage of error is produced by the movement of the Polebot to conclude whether Proportional control is correctly used. The result of the trajectory tracking control carried out is that the Polebot can move on the sine trajectory with an average error on the X axis is 0.02% and on the Y axis 0.11% , on eight trajectories with an error on the X axis 0.06% (backslash percent) and the Y axis 0.02%, and on the circle trajectory with an error on the X axis 0.07% and the Y axis 0.02%. It can be concluded that the Proportional control on the Polebot is quite well implemented.

Keywords: Educational Robot, Omnidirectional Mobile Robot, Robot Operating System, Root Locus, Trajectory Tracking.

1 Introduction

In this day mobile robots have many benefits. for example, in the fields of security, transportation, education etc. [1]. Many studies have been done on educational robots. The purpose of this study of the educational robot is for improve analytical, as a training kit etc. [2]. In this case there are several aspects that need to be considered, that is cost, the platform that is used by the robot, and ease of access for future research [3]. For the application of educational field, the educational mobile robot is designed [4]. Polebot (Polman Open Platform Educational Robot) is an educational mobile robot with a hexagon shape, and it is TWOMR (Three Wheeled Omnidirectional Mobile Robot) [5]. There are several problems with mobile robots. But the Mobile robot has two main problems, that is Path planning and trajectory tracking [6]. The purpose of trajectory

M. U. H. Al Rasyid and M. R. Mufid (eds.), *Proceedings of the International Conference on Applied Science and Technology on Engineering Science 2023 (iCAST-ES 2023)*, Advances in Engineering Research 230,
https://doi.org/10.2991/978-94-6463-364-1_51

tracking is to determine the linear and angular velocity for tracking the trajectory [7]. Considering that movement stability is the important thing of the mobile robot [8], the study of trajectory tracking control needs to be done. There are many control methods that are implemented in wheeled mobile robots [9] it is Sliding Mode Control [10], Model Predictive Control [11], PID [12], etc. In previous study trajectory tracking using PID control method has been implemented to several types of robots. That is Differential Drive mobile robot [13], Three Omnidirectional Mobile robot [14] and four Omnidirectional Mobile robots [15]. The previous study using differential drive mobile robot using PID controller has been done to tracking the NURBS track with small errors [16]. The previous study about four Omnidirectional Mobile robots using PID controller had result 2% linear velocity error [17]. The PI [18] and PID [19] controller has been implemented in the study about trajectory tracking control using Three Omnidirectional Mobile Robot. The study by using PI controller to track the trajectory works well with a closed loop control system [18]. The study of omnidirectional mobile robot to serve the patient in the hospital using P, PID and PD controller is the robot are still unable to reduce error on uneven ground. The PID controller that implemented is better than other [19]. From the result of implemented the P, PI and PID controller from previous study, this study uses Proportional controller to track trajectory with Polebot. A proportional controller was chosen because it has a simple structure and is easy to implement [20]. The communication between microcontroller and personal computer is using ROS (robot operating system) [21] [22]. This study is expected to develop Polman's educational robot.

2 Method

2.1 Omnidirectional Mobile Robot

Omnidirectional mobile robot is a holonomic type of robot [5]. The robot can move in all directions without orientation [10].

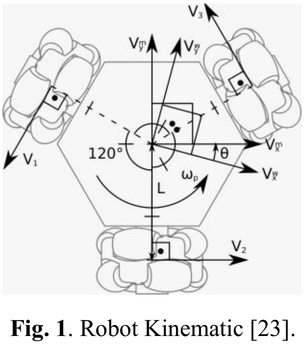


Fig. 1. Robot Kinematic [23].

forward kinematic formula

$$V\,m\,x = 2V2 - V1 - V\,3\,3 \quad (1)$$

$$V\,m\,y = \sqrt{}\,3V3 - \sqrt{}\,3V1\,3 \quad (2)$$

$$\omega p = V1 + V2 + V3\,3L \quad (3)$$

and invers kinematic

$$V1 = -\,V\,m\,x\,2 - \sqrt{}\,3V\,m\,y\,2 + L\omega p \quad (4)$$

$$V2 = V\,m\,x + L\omega p \quad (5)$$

$$V3 = -\,V\,m\,x\,2 + \sqrt{}\,3V\,m\,y\,2 + L\omega p \quad (6)$$

2.2 VDI2206

In this study, VDI2206 solved the problem of mechatronics systems. The problem-solving process is divided into the following, 1) Requirement The first step is problem analysis of the plant to determine the way to solve the problem. 2) System Design At this stage the sub system design for the plant 3) System Integration and Validation The result of the design should be validated using software. 4) Modelling and Analysis Calculation modelling is an approach to determine the characteristics of the plant. 5) Product The final stage is the product.

2.3 System Overview

An overview of the system designed in this study can be seen in the following figure.

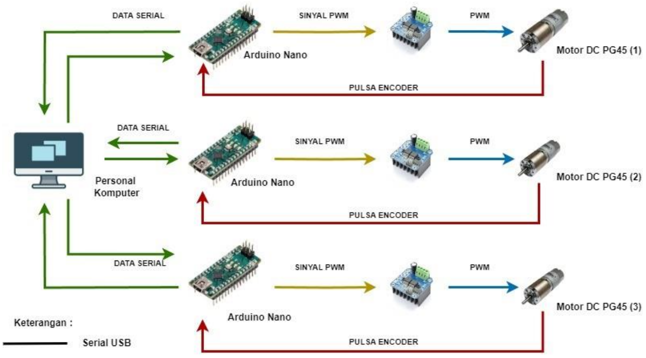


Fig. 2. System Design.

2.4 Mechanical Design

The mechanical design below is the new design of Polebot.

Fig. 3. Polebot Beta Mechanical Design.

2.5 Electrical Design

The Arduino is used to control each motor of the robot.

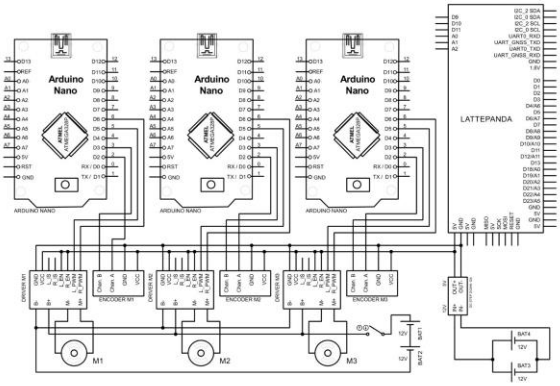


Fig. 4. Polebot Beta Electrical Design.

2.6 Informatic Design

In this study, ROS is the platform for communication for each Arduino nano and personal computer. The following is a programming flowchart of the robot.

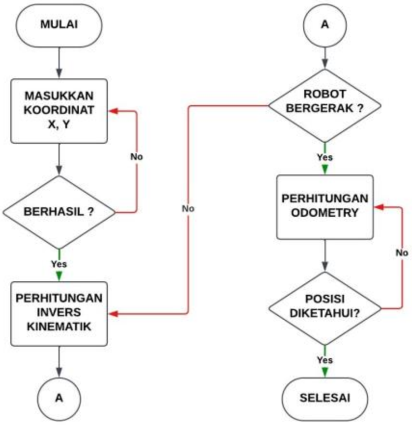


Fig. 5. Flowchart Program.

2.7 Control Design

In this study robots need to be controlled. To determine the controller parameter, we use the root locus tuning method. To determine the transfer function of the DC motor, system identification is used from matlab. By using the PWM input and RPM output it can determine the transfer function of the DC motor.

Transfer function of the robot uses form the previous study about soccer robot with the following equation, [24]

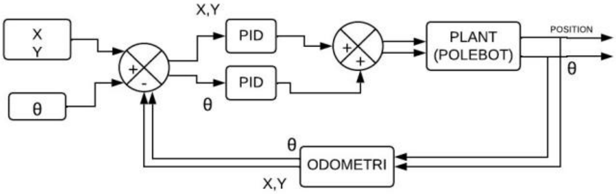


Fig. 6. Closed Loop Control System Polebot.

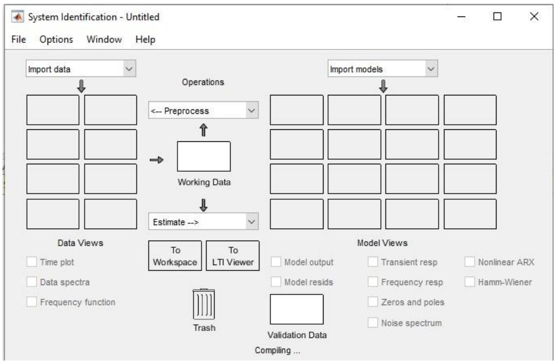


Fig. 7. System Identification Matlab.

Hp(s) = X(s) U(S) = √ 3α ms2 + 3β(s) 2 (7)

α = motor torque constant β = motor speed constant m = mass of the robot transfer function of the robot can be written as,

Hp(s) = X(s) U(S) = 0.011768 0.5s 2 + 0.72s 2 (8)

2.8 Trajectory Tracking

The robot will move to follow the calculated trajectory. To analyze the movement, the robot will move to sine trajectory, eight trajectory and circle trajectory 9 – 11 as follows [25].

for t = 0 ≤ t ≤ 240

$$\begin{bmatrix} V_x \\ V_y \\ \theta \end{bmatrix} = \begin{bmatrix} 0.016 * t \\ sin(1.5 * t) \\ 0 \end{bmatrix} \qquad (9)$$

for t = 0 ≤ t ≤ 390

$$\begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = \begin{bmatrix} cos\, t \\ \frac{sin\, 2t}{2} \\ 0 \end{bmatrix} \qquad (10)$$

for t = 0 ≤ t ≤ 390

$$\begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = \begin{bmatrix} \cos t \\ \sin t \\ 0 \end{bmatrix} \tag{11}$$

3 Results

3.1 Mechanical Design Implementation

The following is an implementation of the mechanical design.

Fig. 8. Mechanical Design Implementation.

3.2 Electrical Design Implementation

Arduino is installed and applied to every DC motor of the robot.

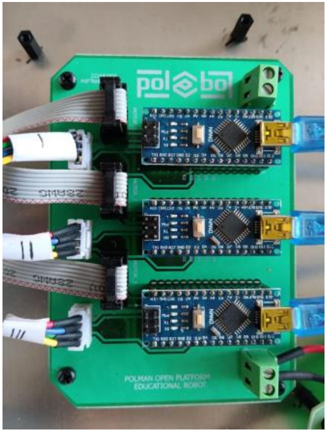


Fig. 9. Electrical Wiring Implementation.

3.3 Information Design Implementation

The interface of the robot is using ubuntu’s terminal to drive the robot as follows.

```
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://miftah-hp:38289/
ros_comm version 1.15.15


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.15

NODES

auto-starting new master
process[master]: started with pid [4889]
ROS_MASTER_URI=http://miftah-hp:11311/

setting /run_id to 7ee572a8-38a8-11ee-8ae7-b5fc300b37f3
process[rosout-1]: started with pid [4902]
started core service [/rosout]
```

```
miftah@miftah-hp:~/polebot_beta$ rosrun polebot1 input.py
coordinate_X     :  0
coordinate_Y     :  1
coordinate_Theta :  0
```

```
X =  [0.0]
Y =  [0.0]
W =  [0.0]
Posisi_X =  [0.0]
Posisi_Y =  [0.0]
Omega    =  [0.0]
X =  [0.0]
Y =  [0.0]
W =  [0.0]
Posisi_X =  [0.0]
Posisi_Y =  [0.0]
Omega    =  [0.0]
X =  [0.0]
Y =  [0.0]
W =  [0.0]
Posisi_X =  [0.0]
Posisi_Y =  [0.0]
Omega    =  [0.0]
X =  [0.0]
Y =  [0.0]
W =  [0.0]
Posisi_X =  [0.0]
Posisi_Y =  [0.0]
Omega    =  [0.0]
```

Fig. 10. Interface.

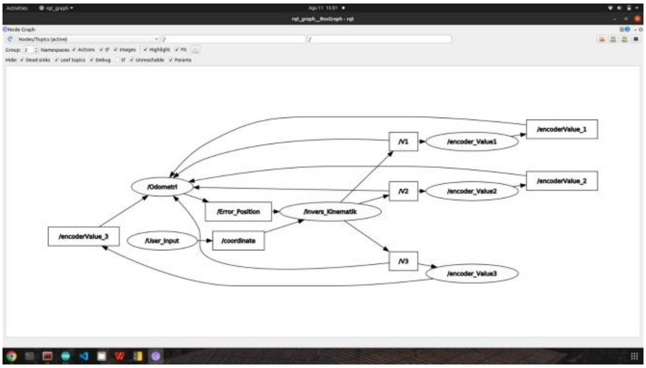
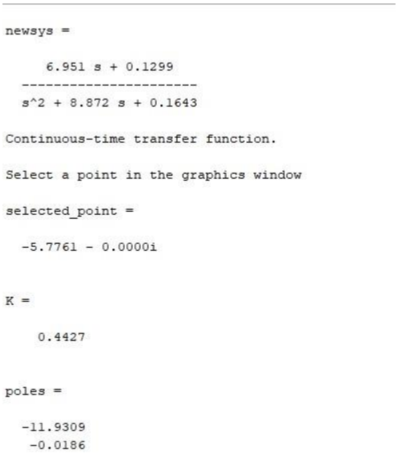


Fig. 11. Rqt graph.

3.4 Result of Determine of PID Control Parameters on Motors and Robot

1) Motor Control By using system identification, the transfer function of each motor as follows,



```
newsys =

     6.951 s + 0.1299
  ----------------------
  s^2 + 8.872 s + 0.1643

Continuous-time transfer function.

Select a point in the graphics window

selected_point =

  -5.7761 - 0.0000i


K =

    0.4427


poles =

  -11.9309
   -0.0186
```

Fig. 12. Motor 1 response after using PID.

Motor 1 response with a rise time of 0.6 seconds with an overshot 0.24% and a settling time of 1.09 seconds.

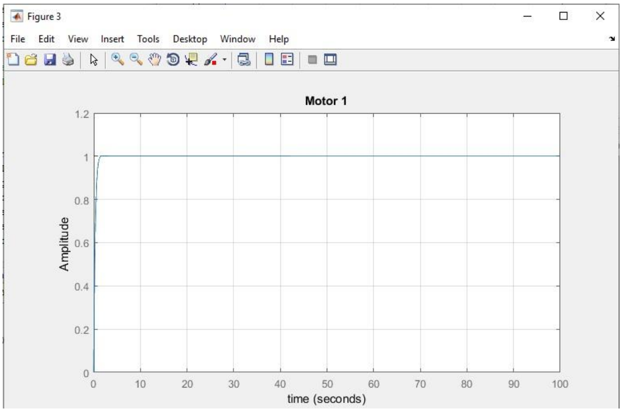
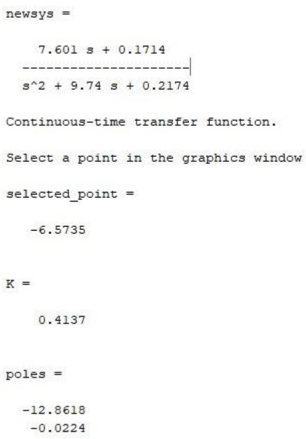


Fig. 13. Motor 1 Transfer Function.



```
newsys =

    7.601 s + 0.1714
  ---------------------
  s^2 + 9.74 s + 0.2174

Continuous-time transfer function.

Select a point in the graphics window

selected_point =

   -6.5735


K =

    0.4137


poles =

  -12.8618
   -0.0224
```

Fig. 14. Motor 2 response after using PID.

Motor 2 response with a rise time of 0.6 seconds with an overshoot 0.03% and a settling time of 1.23 seconds.

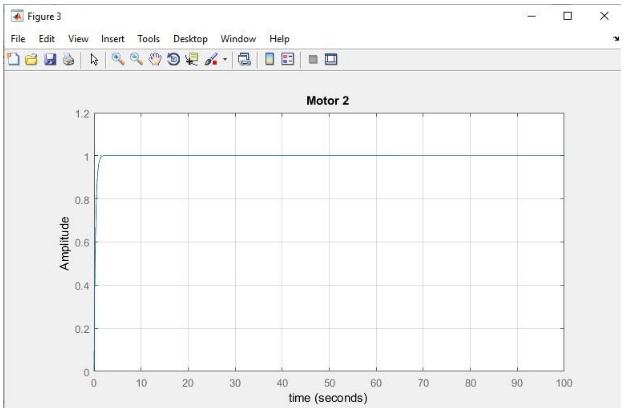
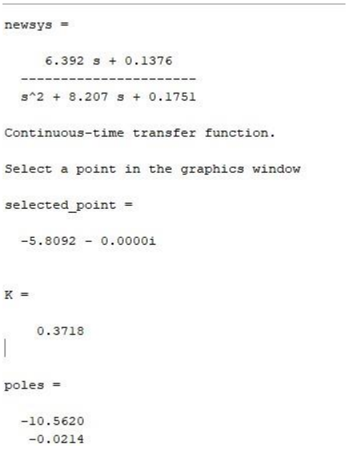


Fig. 15. Motor 2 Transfer Function.

The motor 3 response with a rise time of 0.6 seconds with an overshoot 0.24% and a settling time of 1.09 seconds



```
newsys =

     6.392 s + 0.1376
  ----------------------
  s^2 + 8.207 s + 0.1751

Continuous-time transfer function.

Select a point in the graphics window

selected_point =

  -5.8092 - 0.0000i


K =

    0.3718

|
poles =

  -10.5620
   -0.0214
```

Fig. 16. Motor 3 response after using PID.

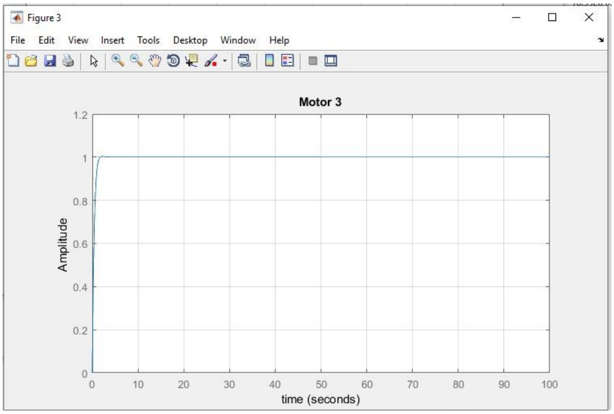


Fig. 17. Motor 3 Transfer Function

2) Robot Control Using root locus tuning method, proportional control is obtained as follows,

3.5 Trajectory Tracking Result

In this study the robot will be track trajectory of sine, eight trajectory and circle trajectory following result,

Table 1. Result Of The Sine Trajectory Tracking.

	Input	Experiment	Error

t	X	Y	X	Y	Error X	Error Y
0	0.000	0.000	0.000	0.000	0.000	0.000
1	0.016	0.026	0.003	0.001	0.013	0.025
2	0.032	0.052	0.011	0.007	0.021	0.045
3	0.048	0.078	0.023	0.018	0.025	0.060
4	0.064	0.105	0.037	0.036	0.027	0.069
5	0.080	0.131	0.054	0.055	0.026	0.076
6	0.096	0.156	0.070	0.076	0.026	0.080
7	0.112	0.182	0.088	0.097	0.024	0.085
8	0.128	0.208	0.105	0.117	0.023	0.091
9	0.144	0.233	0.124	0.137	0.020	0.096
10	0.160	0.259	0.142	0.158	0.018	0.101
...	...	...	...	...	...	...
239	3.824	- 0.026	3.775	- 0.191	0.049	0.165

The error is 0.02% on the X coordinate and 0.11% on the Y coordinate. With the tracking chat as follows

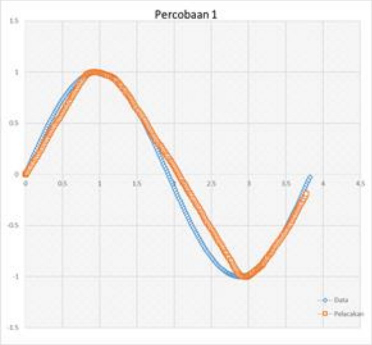


Fig. 18. Tracking results of sinus trajectory.

Table II. Result Of The Eight Trajectory Tracking.

	Input		Experiment		Error	
t	X	Y	X	Y	Error X	Error Y
0	1.000	0.000	0.000	0.000	1.000	0.000
1	1.000	0.017	0.003	0.003	0.997	0.014
2	0.999	0.035	0.020	0.011	0.979	0.024
3	0.999	0.052	0.045	0.020	0.954	0.032
4	0.998	0.070	0.073	0.034	0.925	0.036
5	0.996	0.087	0.104	0.049	0.892	0.038
6	0.995	0.104	0.132	0.066	0.863	0.038
7	0.993	0.121	0.162	0.083	0.831	0.038
8	0.990	0.138	0.191	0.099	0.799	0.039
9	0.988	0.155	0.223	0.116	0.765	0.039
10	0.985	0.171	0.254	0.134	0.731	0.037
...	....	....	....	....	....	....

389	0.875	0.424	0.912	0.376	0.037	0.048

The error is 0.07% on the X coordinate and 0.03% on the Y coordinate. With the tracking chat as follows

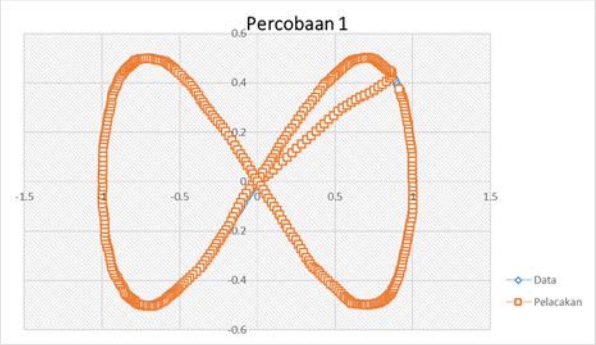


Fig. 19. Tracking results of eight trajectory.

4 Conclusion

The result of this study, the robot can successfully move tracking the trajectory with the average error as follow.

Table III. Result Of The Circle Trajectory Tracking.

	Input		Experiment		Error	
t	X	Y	X	Y	Error X	Error Y
0	1	0.000	0.000	0.000	1.000	0.000
1	1.000	0.017	0.006	0.003	0.994	0.014
2	0.999	0.035	0.027	0.011	0.972	0.024
3	0.999	0.052	0.052	0.024	0.947	0.028
4	0.998	0.070	0.082	0.037	0.916	0.033
5	0.996	0.087	0.113	0.051	0.883	0.036
6	0.995	0.105	0.144	0.069	0.851	0.036
7	0.993	0.122	0.172	0.087	0.821	0.035
8	0.990	0.139	0.202	0.104	0.788	0.035
9	0.988	0.156	0.233	0.122	0.755	0.034
10	0.985	0.174	0.266	0.140	0.719	0.034
....	....	....	....	....	....	....
389	0.875	0.485	0.901	0.428	0.023	0.022

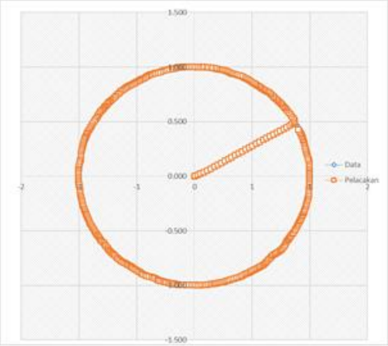


Fig. 20. Tracking results of circle trajectory

1) The sine trajectory error is 0.02% in X coordinate and 0.11% in Y coordinate.
2) The eight-trajectory error is 0.06% in X coordinate and 0.02% in Y coordinate.
3) The circle trajectory error is 0.07% in X coordinate and 0.02% in Y coordinate.

Acknowledgement

This work has been supported by PT Seascape Surveys Indonesia and Internal research funding from Bandung Polytechnic for Manufacturing.

References

1 H. Yang, S. Wang, Z. Zuo, & P. Li, "Trajectory tracking for a wheeled mobile robot with an omnidirectional wheel on uneven ground," IET Control Theory & Applications, 14(7), 921-929, (2020).

2 A. Erdogmus, B. Yarıs¸, F. Vatansever, Y. Umur, M. Tas¸, H. Yavuz, "Development of Educational Robot & User Interfaces for Robotic Applications," ICENTE21 - International Conference on Engineering Technologies, November, (2021).

3 F. Arvin, J. Espinosa, B. Bird, A. West, S. Watson, & B. Lennox, "Mona: an affordable open-source mobile robot for education & research," Journal of Intelligent and Robotic Systems, 94, 761-775, (2019).

4 T. Guo, M. Li, Y. Han, G. Lu, T. Qie, & Q. Zhang, "Design of educational mobile robot," In 2020 Chinese Automation Congress (CAC) (pp. 3234-3238) IEEE, November, (2020).

5 N. Hacene & B. Mendil, "Motion analysis and control of three-wheeled omnidirectional mobile robot," Journal of Control, Automation and Electrical Systems, 30, 194-213, (2019).

6 O. Martins, A. Adekunle, S. Adejuyigbe, O. Adeyemi, & M. Arowolo, "Wheeled Mobile Robot Path Planning and Path Tracking Controller Algorithms: A Review," Journal of Engineering Science & Technology Review, 13(3), (2020).

7 N. Uddin, "Trajectory tracking control system design for autonomous two-wheeled robot," Jurnal Infotel, 10(3), 90-97, (2018).

8 F. Rubio, F. Valero, and C. Llopis-Albert, “A review of mobile robots: Concepts, methods, theoretical framework, & applications,” International Journal of Advanced Robotic Systems, 16(2), 1729881419839596,(2019).

9 [9] O. Access, “A Sliding mode control method for trajectory tracking control of wheeled mobile robot,” Journal of Physics: Conference Series, Volume 1074, The International Conference on Mechanical, Electric and Industrial Engineering (MEIE2018) 26–28, Hangzhou, China, May, (2018).

10 E. M. Ijaabo, A. Alsharkawi, & A. R. Firdaus, “Trajectory tracking of an omnidirectional mobile robot using sliding mode control,” 2nd International Conference on Applied Engineering (ICAE) (pp. 1-6), IEEE, October, (2019).

11 C. Wang, X. Liu, X. Yang, F. Hu, A. Jiang, & C. Yang, “Trajectory tracking of an omni-directional wheeled mobile robot using a model predictive control strategy,” Applied Sciences, 8(2), 231, (2018).

12 R. P. Borase, D. K. Maghade, S. Y. Sondkar, & S. N. Pawar, “A review of PID control, tuning methods & applications,” International Journal of Dynamics & Control, 9, 818-827, (2021).

13 M. Gheisarnejad, & M. H. Khooban, “Supervised control strategy in trajectory tracking for a wheeled mobile robot,” IET Collaborative Intelligent Manufacturing, 1(1), 3-9, (2019).

14 A. Andreev, O. Peregudova, & K. Sutyrkina, “Robust Trajectory Tracking Control of Omni-Wheeled Mobile Robots,” International Journal of Control Systems & Robotics, Volume 4, (2019).

15 K. D. H. Thi, M. C. Nguyen, H. T. Vo, D. D. Nguyen, & A. D. Bui, “Trajectory tracking control for four-wheeled omnidirectional mobile robot using Backstepping technique aggregated with sliding mode control,” First International Symposium on Instrumentation, Control, Artificial Intelligence, & Robotics (ICA-SYMP) (pp. 131-134). IEEE, January, (2019).

16 N. H. Thai, T. T. K. Ly, H. Thien, & L. Q. Dzung, “Trajectory tracking control for differential-drive mobile robot by a variable parameter PID controller,” International Journal of Mechanical Engineering & Robotics Research, 11(8), 614-621, (2022).

17 A. Ma’arif, N. M. Raharja, G. Supangkat, F. Arofiati, R. Sekhar, and D. U. Rijalusalam, “Pid-based with odometry for trajectory tracking control on four-wheel omnidirectional covid-19 aromatherapy robot,” Emerging Science Journal, 5, 157-181, (2021).

18 M. A. Kawtharani, V. Fakhari, & M. R. Haghjoo, “Tracking Control of an Omni-Directional Mobile Robot,” International Congress on HumanComputer Interaction, Optimization & Robotic Applications (HORA) (pp. 1-8). IEEE, June, (2020).

19 S¸. Yildirim, “Trajectory Control of Designed Experimental Mobile Robot,” Recent Innovations in Mechatronics, 7(1.), 1-5, (2020).

20 A. Ma’arif, R. Istiarno, & S. SUNARDI, “Kontrol Proporsional Integral Derivatif (PID) pada Kecepatan Sudut Motor DC dengan Pemodelan Identifikasi Sistem dan Tuning,” ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, and Teknik Elektronika, 9(2), 374, (2021).

21 P. Anggraeni, M. Mrabet, M. Defoort, & M. Djemai, “Development of a wireless communication platform for multiple-mobile robots using ROS,” 6th International Conference on Control Engineering & Information Technology (CEIT) (pp. 1-6). IEEE, October, (2018).

22 M. Marian, F. Stˆınga, M. T. Georgescu, H. Roibu, D. Popescu, &˘ F. Manta, “A ROS-based control application for a robotic platform using the gazebo 3D simulator,” 21th International Carpathian Control Conference (ICCC) (pp. 1-5). IEEE, October, (2020).

23 GuiRitter, “OpenBase.” https://github.com/GuiRitter/OpenBase (accessed Aug. 09, 2023)(2023).

24 M. DaneshPanah, A. Abdollahi, H. Ostadi, & H. A. Samani, “Comprehensive omni-directional soccer player robots,” Robotic Soccer, IntechOpen, (2007).

25 F. Fahmizal, A. Priyatmoko, & A. Mayub, "Implementasi Kinematika Trajectory Lingkaran pada Robot Roda Mecanum," Jurnal Listrik, Instrumentasi, dan Elektronika Terapan (JuLIET), 3(1), Mei, (2022).