# EXPERIMENTAL IN HEAD TRACKING CONTROL OF A FOUR OMNI WHEELED MOBILE ROBOT SYSTEM

SRUN CHANNARETH*, LONH VANNSITH, NY VIRBORA

*Faculty of Electronics, National Polytechnic Institute of Cambodia, Phnom Penh, Cambodia*

**Corresponding author: nareth16npic@gmail.com*

*(Received: 01 August 2022; Accepted: 11 October 2022; Published on-line: 01 November 2022)*

***ABSTRACT:*** This article describes an experimental investigation into the head tracking and control of a four-wheeled Omni-Robot. This study described the control scheme of the omnidirectional mobile robot. Additionally, the kinematics of the mobile robot is presented., A mobile robot with four Omni wheels was also developed on real hardware to verify the controller architecture design. The robot system is modeled and then implemented with the Arduino Microcontroller. This robot has two rotary encoders for running on the x-axis and y-axis, respectively. The gyro sensor (MPU-6050) is placed in the robot's middle and tracks data obtained by yaw angle. MATLAB Simulink was used to find the PD controller based on manual adjustment. The control architecture was deployed to complete and run the experiment on the mobile robot to evaluate the outcomes. The experiment results show the robot's motion control improvement in various scenarios.

*KEY WORDS: Head Control, Holonomic Drive, Kinematic, PD Controller*

## 1. INTRODUCTION

A mobile robot that uses an Omni-wheel or holonomic drive can traverse through tight spaces since it has the ability to move in any direction. The omnidirectional mobile robot has been widely used in various applications, such as autonomous ground vehicles (AGV), service robots inside settings, and production line special motion maneuvers in limited spaces. The terrain type only limits the use case and the maximum weight its wheels can support.

Omni wheels have tiny discs all the way around, perpendicular to the rolling direction. The result is that the wheel will roll vigorously while also easily sliding laterally. A control system is required to regulate the speed and direction of rotation of each wheel to move the vehicle in the desired direction. The kinematic equation has been developed and well-documented during the past year, as seen in the work displayed in [1-2]. The size and the actual configuration of the robot, such as whether it has three or four wheels, must be used to specifically calculate the kinematics of the robot that must be managed.

The control problems of motion and regulation are important things to consider. One common challenge of control is to keep track of the heading angle and control it so we can reduce the drifting of the robot over time due to the accumulated error from the encoder and the drifting due to the terrain condition [3-9].

In earlier studies, such as those covered in [10–13], to try to solve these problems, the Trajectory Linearization Control (TLC) approach and Optimal Trajectory Tracking Control (OTRC) were used. The "trajectory linearization control" approach is based on linearization along a notional trajectory (TLC). The TLC successfully applies to missile and reusable launchvehicle-flight-control systems by fusing nonlinear dynamic inversion and linear time-varying eigenstructure assignment.

In this paper, we try to tackle the problem by using a technique of heading angle control with the help of YAW angle measurement. We use the IMU sensor to get the YAW angle. We decided to use Simulink for the coding and calculation system. This paper is divided into five sections. Section 1 is an introduction, and Section 2 is kinematics. Section 3 is about controller design. Section 4 discusses the experiment and the result, and Section 5 discusses the conclusion and future work.

## 2. KINEMATICS

One type of robot employed in many fields is the 4WD holonomic drive. They have four wheels that can move in any direction at any moment and are omnidirectional [1]. The robot and its wheel positions are shown in figure 1.

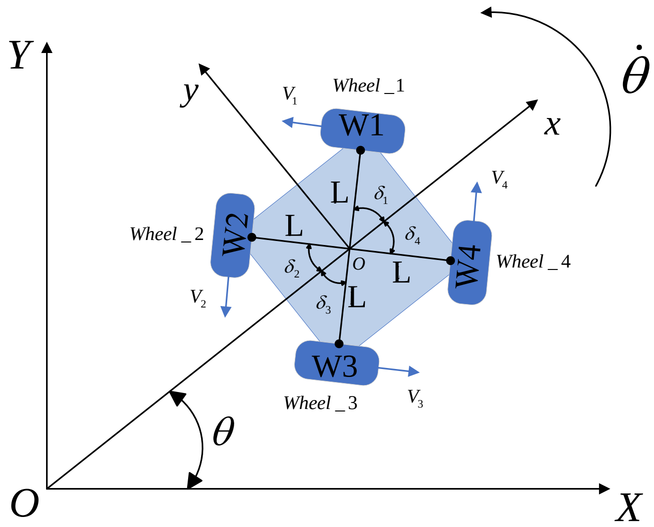


Fig. 1. The wheel position of the robot in OXY

Where XOY is the world coordinate frame of the robot. The moving direction of the robot is W. The δ is the angle between the wheels. The V is the velocity of each wheel, and its positive direction is anticlockwise. L is the distance between the middle of the robot body of every wheel. The robot position and orientation within the robot coordinate frame are expressed as $X_m = (x\ y\ \theta)^T$, and the robot posture is presented within the world coordinate frame as $q = (X\ Y\ \theta)^T$ the link between $X_m$ and q is followed in Formula (1):

$$X_m = R(\theta)\cdot q = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}\cdot q \quad (1)$$

The kinematic model of the robot can be expressed as Formula (2):

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{pmatrix} = \begin{pmatrix} -\sin(\delta_1+\theta) & +\cos(\delta_1+\theta) & L \\ -\sin(\delta_2-\theta) & -\cos(\delta_2-\theta) & L \\ +\sin(\delta_2+\theta) & -\cos(\delta_2+\theta) & L \\ +\sin(\delta_1-\theta) & +\cos(\delta_1-\theta) & L \end{pmatrix}\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{pmatrix} = J\dot{q} \quad (2)$$

https://doi.org/10.51630/ijes.v3i3.67

# 3. SYSTEM DESIGN

## 3.1. Block Diagram of the Robot System

Fig. 2 shows the block diagram of the robot system; the system consists of an emergency stop button, remote control, encoder, MPU-6050, and a touch screen for the input part. The output parts are the LED indicator, buzzer, and smart PID drivers. The robot can receive commands in two ways. The first is from the remote control, and the second is from the touchscreen panel.

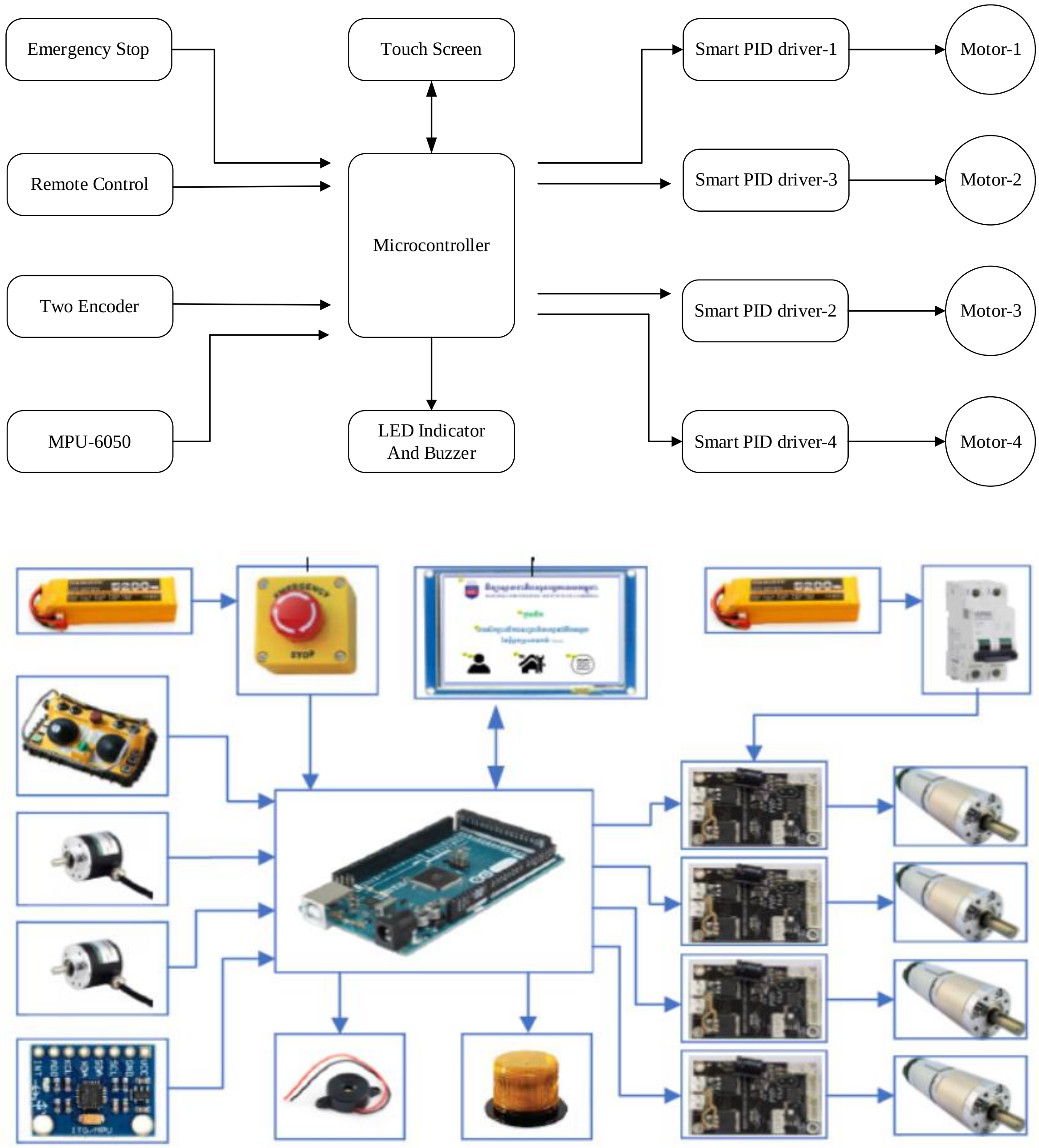


Fig. 2. The block diagram of the robot system.

https://doi.org/10.51630/ijes.v3i3.67

### 3.2. Controller Design

One of the most well-known controllers in the control system is PID control. A four-wheel robot's head tracking and control utilize a portion of PID, which controls the PD. In this application, head tracking takes advantage of it. Because the controller doesn't have to match the output to the input at every step, it performs better. Fig. 3 shows the structure of the PD controller.

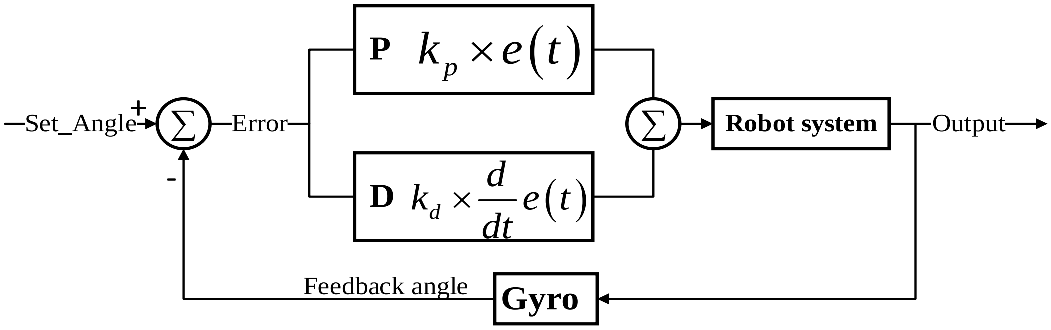


Fig. 3. The PD Control Structure.

The proportional (P) part adjusts the output according to the input. As can be seen, we need an error between output and input for this system. The error is multiplied by the gain value, Kp. The D part tries to predict the future of the control action; it can be considered the damp part. Setting D will help the system avoid overshoot, thus resulting in smoother operation.

We can find the Kp and Kd by manually tuning between the mobile robot and the Simulink. In MATLAB Simulink, we designed the control block for the robot system. The hardware support package communicates between the block and the robot. In the first step, Kp and Kd gain is found by increasing Kp and keeping Kd to zero. The second step is to increase Kp until the response is not too much overshoot, then increase Kd to decrease overshoot.

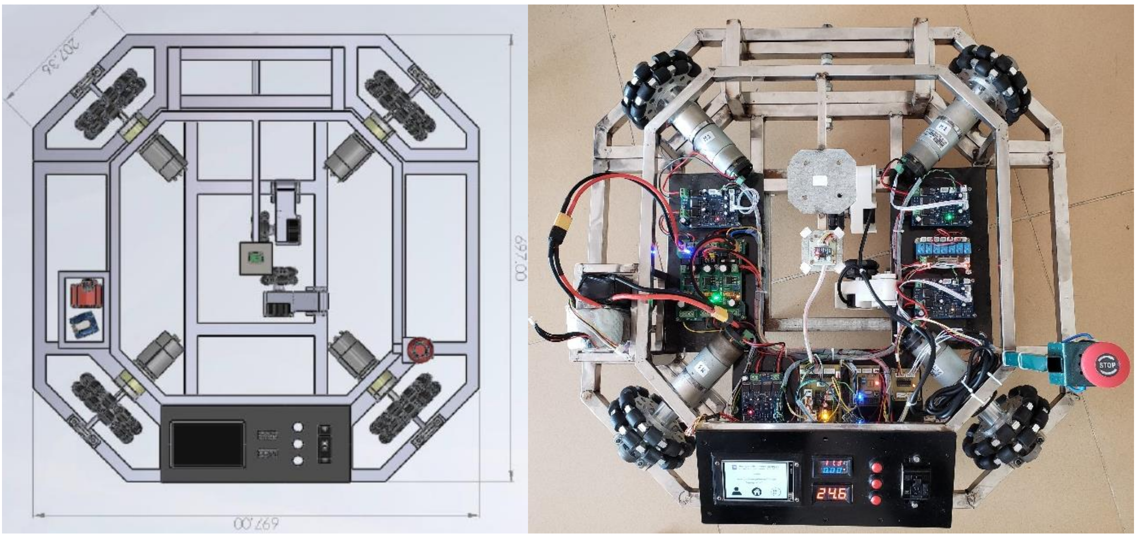

Fig. 4. The Robot's Structure Design.

### 3.3. Omni Mobile Robot Design

This robot has a length of 697mm, a width of 697mm, a height of 100mm, and a weight of 15kg, as shown in Fig. 4. A chamfer length of 207.36mm at each corner reduces the collision angle, which can cause harm.

The bottom layout of the robot is equipped with two rotary encoders connected with small Omni wheels, as shown in Fig. 5. Fig. 6 shows the position of the gyro sensor. This sensor is put in the middle of the robot body.

The four Omni wheels are connected to the DC motor via a hub. Position the four corners between the wheels and the other wheels of equal length relative to the center of the robot, each wheel as shown in Fig. 7.

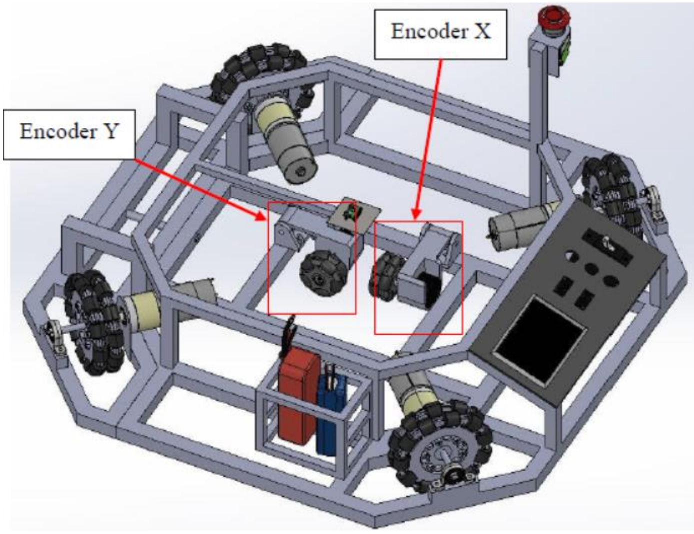


Fig. 5. The Positioning and Direction of the Rotary Encoder.

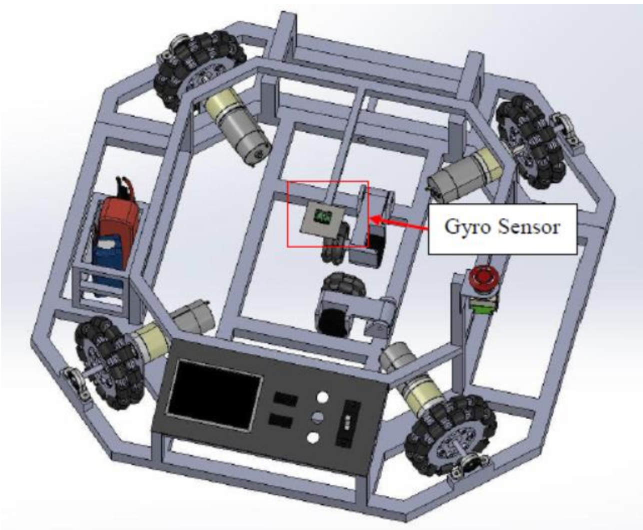


Fig. 6. The Place of Gyro Sensor.

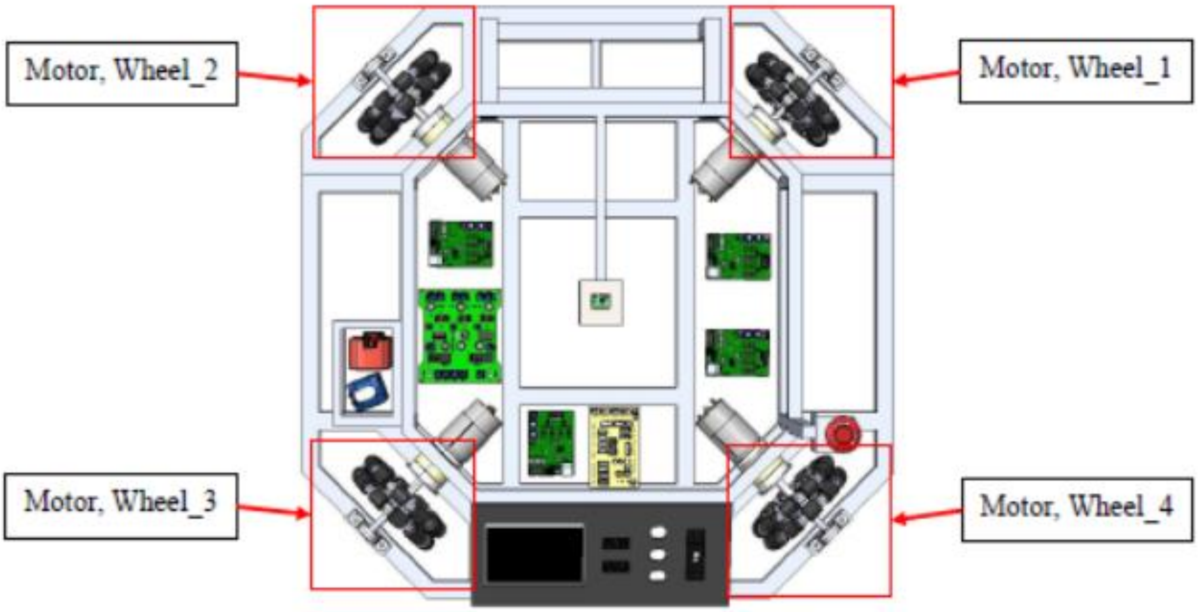


Fig. 7. The Omni Wheels Coupled with DC motors.

### 3.4. Robot Controller Design with MATLAB Simulink

The block shown in Fig. 8 is designed with MATLAB Simulink. We can determine the robot's position by entering the value into Set_Position_X and Set_Position_Y. For set_angle is set the angle of the robot's body.

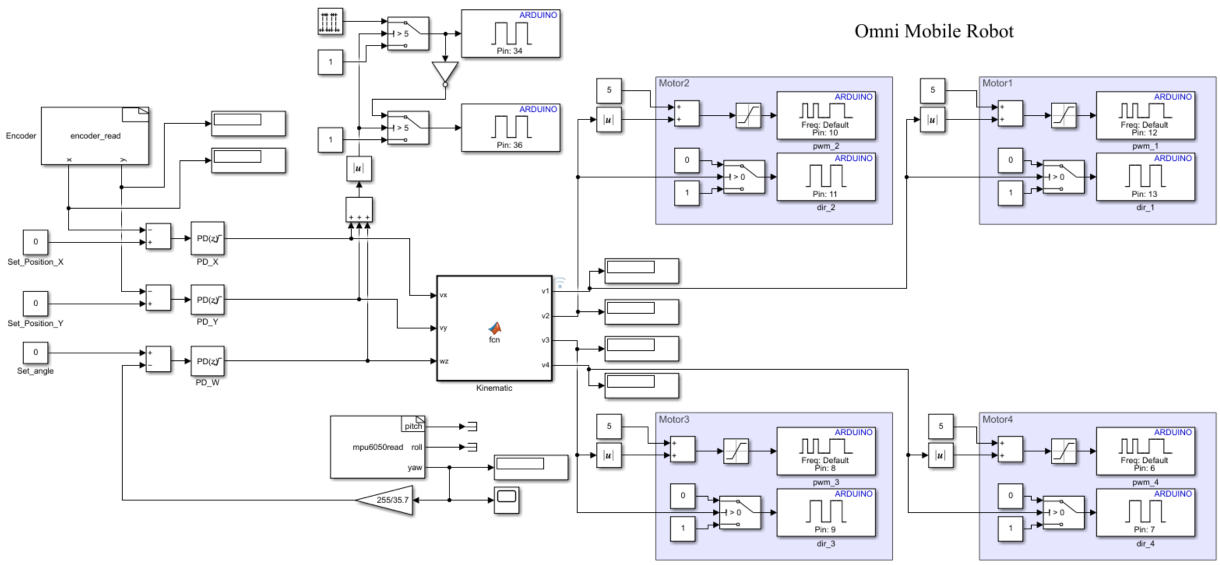


Fig. 8. MATLAB Simulink Control Block.

https://doi.org/10.51630/ijes.v3i3.67

## 4. EXPERIMENTAL AND RESULT

### 4.1. Testing the robot moves on the X-axis without and with head control

Fig. 9 shows the testing robot on a trajectory following the X-axis to experiment without and with a tolerance of the head tracking control on this axis. The distance is 3m from the start to the end. The time measurement takes 10 seconds to complete. Fig. 10 and 11 show the result of the set angle and error angle when the robot moves on the x-axis.

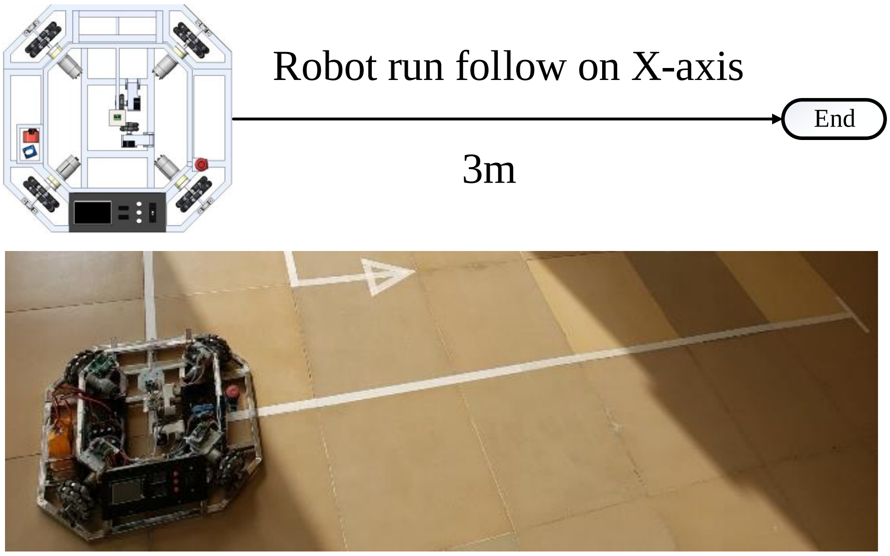


Fig. 9. The robot runs on X-axis.

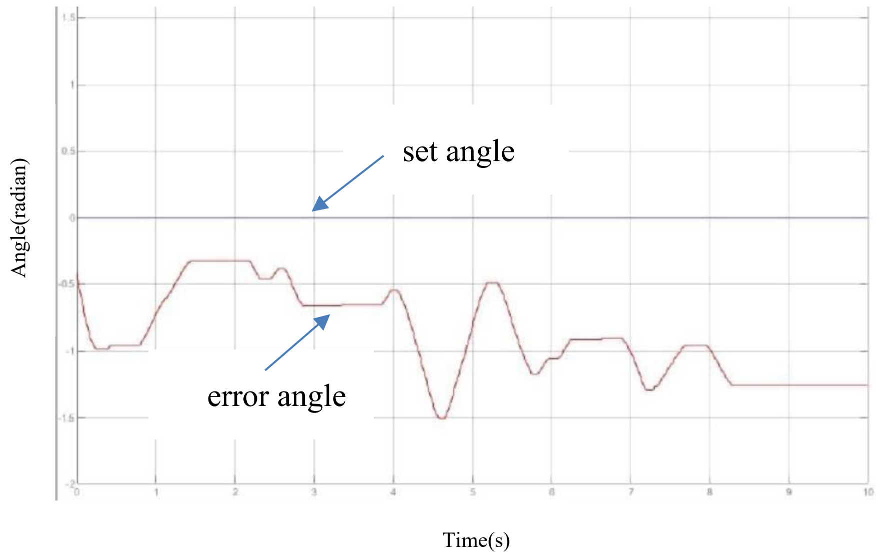


Fig. 10. The robot's head error without using head control runs on the x-axis.

### 4.2. Testing the robot moves on the Y-axis without and with head control

Fig. 12 shows the testing robot on a trajectory following the Y-axis to experiment without and with a tolerance of the head tracking control on this axis. The distance is 3m from the start to the end as the same on X-axis. The time measurement takes 10 seconds to complete. Fig. 13 and 14 show the result of the set angle and error angle when the robot moves on the y-axis.

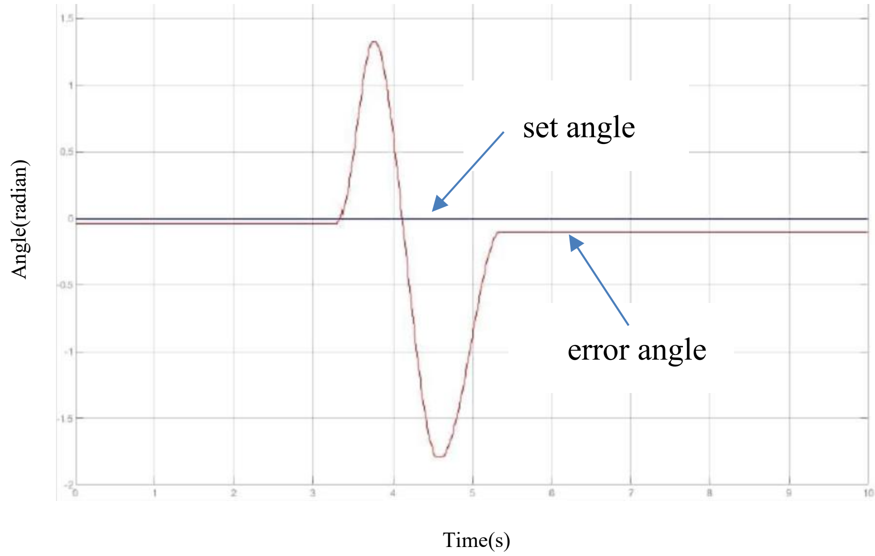


Fig. 11. The robot's head error using head control run on the x-axis.

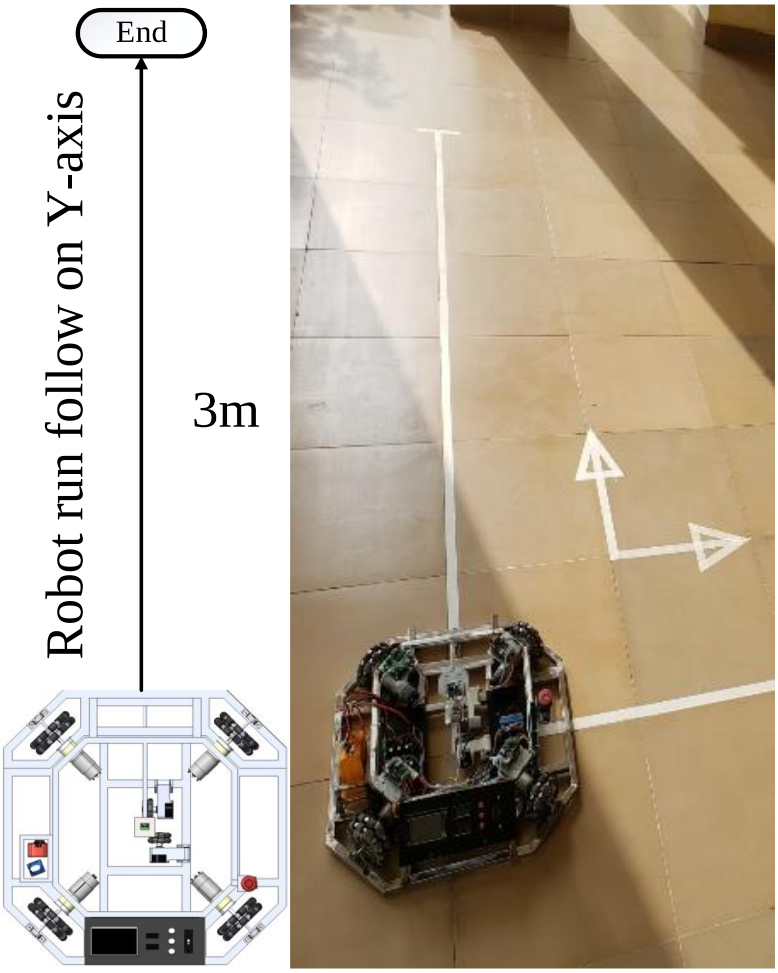


Fig. 12. The robot runs on Y-axis.

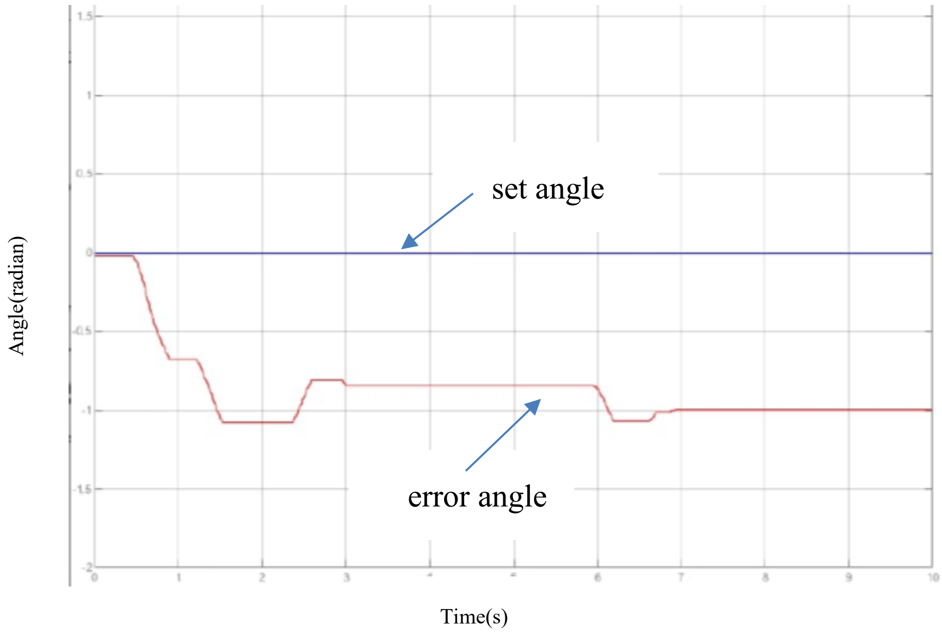


Fig. 13. The robot's head error without using head control run on the Y-axis.

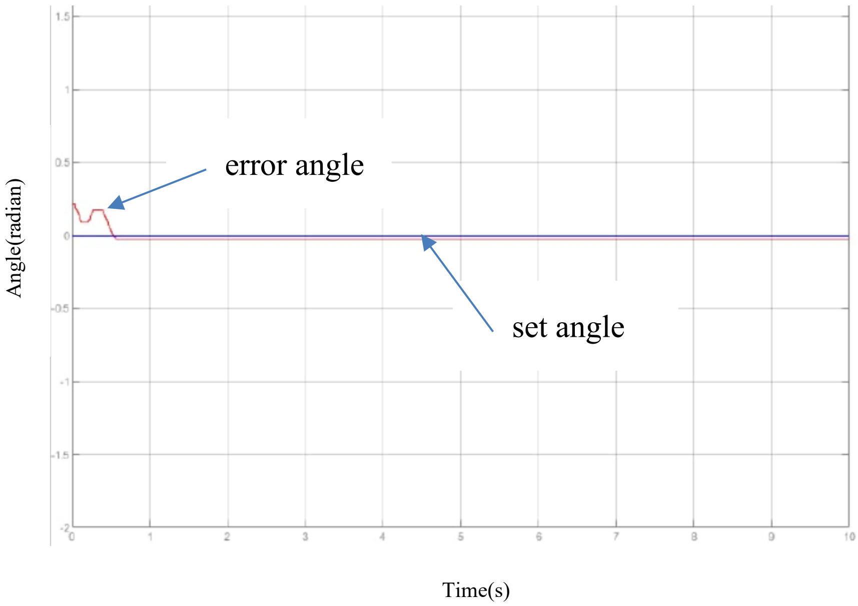


Fig. 14. The robot's head error using head control run on the Y-axis.

### 4.3. Testing the robot moves on a square trajectory without and with head control

A square trajectory is the last test to find any error to make sure the tolerance of the head tracking control is maintained, as shown in Fig. 15. Fig. 16 and 17 show the result of the set angle and error angle when the robot moves on the square trajectory.

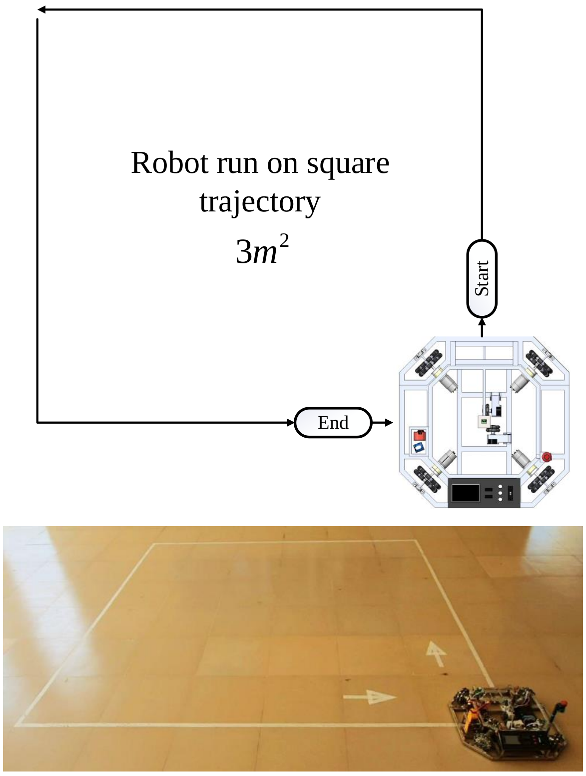


Fig. 15. The robot runs on a square trajectory.

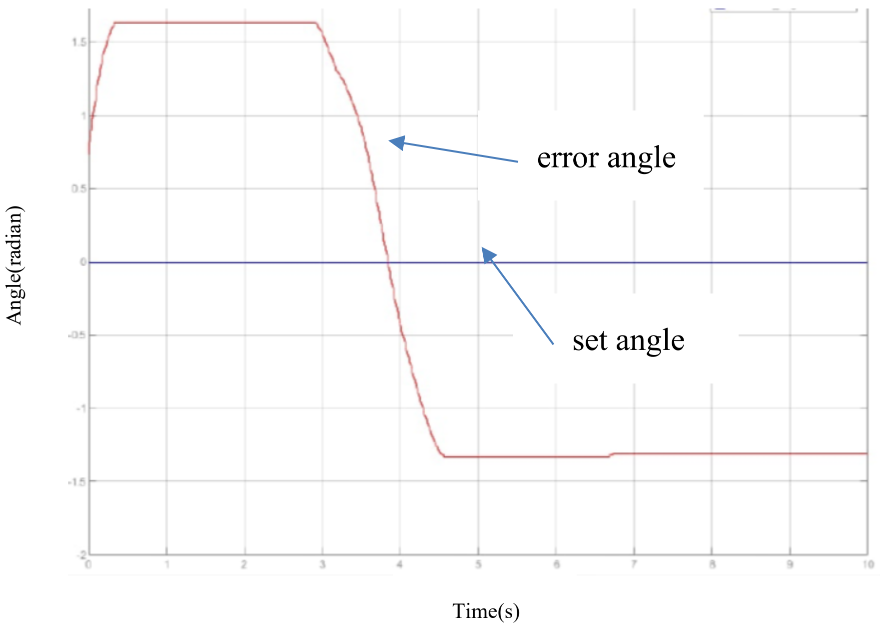


Fig. 16. The robot's head error without using head control runs on the square.

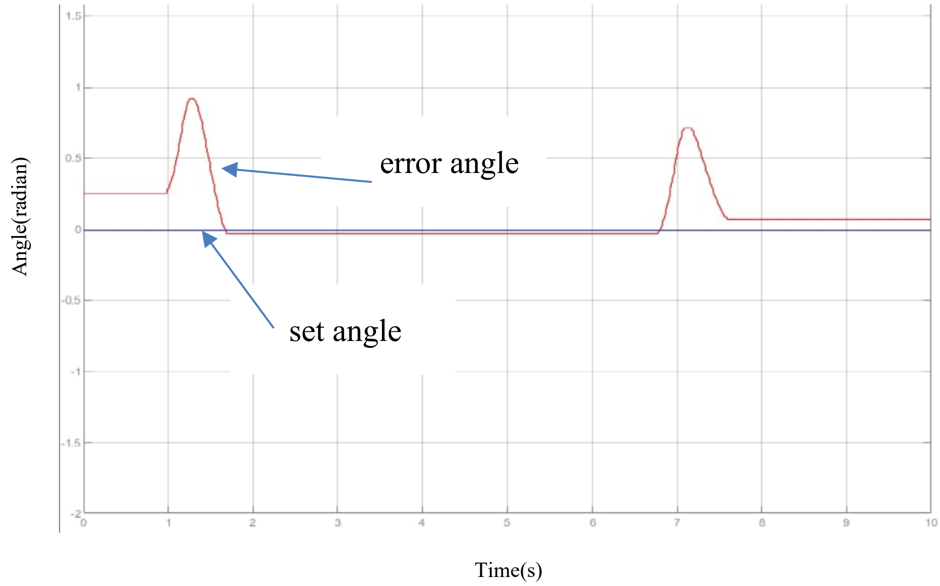


Fig. 17. The robot's head error using head control run on the square.

### 4.4. Result

After the experiment is completed, we get results as shown in table1.

Table 1: The testing result of robot runs other trajectories

| No | Trajectory | Error angle with head tracking (radian) | Error angle with head tracking (degree) | Error angle without head tracking (radian) | Error angle without head tracking (degree) |
|---|---|---|---|---|---|
| 1 | X-axis | 0.1 rad | 5.72958° | 1.25 rad | 71.61972° |
| 2 | Y-axis | 0.05 rad | 2.864789° | 1rad | 57.2958° |
| 3 | Square | 0.1 rad | 5.72958° | 1.35 rad | 77.3493° |

## 5. CONCLUSION

An experimental study is proposed on a four-wheeled robot's control and head tracking. We applied the Holonomic drive's kinematics and reduced the Omni-wheel robot's heading error. It shows that the PD controller is suitable for controlling this system. Two conditions, one without head tracking and the other with head tracking, were used in the experiment. We put each controller to the test using a variety of trajectory configurations, such as running in a square trajectory, along X, along Y, or at an adjacent angle. The experiment results confirmed that the controller with a head tracking angle significantly improved over the one without tracking control. To achieve a precise heading angle for the robot, we aim to explore sensor fusion in more detail in upcoming work.

## REFERENCES

[1] C. J. Wang Jianbin, "An Adaptive Sliding Mode Controller for Four-wheeled Omnidirectional Mobile", in The 31st Chinese Control and Decision Conference, (2019), 5591-5596, DOI: 10.1109/CCDC.2019.8832894.

[2] Fahmizal, C.-H. Kuo, "Trajectory and heading tracking of a mecanum wheeled robot using fuzzy logic control", (2016) International Conference on Instrumentation, Control and Automation (ICA), 54-59, DOI: 10.1109/ICA.2016.7811475.

[3] S.H. Mahboobi, Aria Alasty, "Trajectory Tracking of a Mobile Robot Using Fuzzy Logic Tuned by Genetic Algorithm", International Journal of Engineering-Transaction A: Basics, Vol.19, No. 1, (2006), 95-104.

[4] E. Malayjerdi, H. Kalani and M. Malayjerdi, "Self-Tuning Fuzzy PID Control of a Four-Mecanum Wheel Omni-directional Mobile Platform", 26th Iranian Conference on Electrical Engineering, (2018), 816-820, DOI: 10.1109/ICEE.2018.8472568.

[5] Yong Liu, Xiaofei Wu, J. Jim Zhu and Jae Lew, "Omni-directional mobile robot controller design by trajectory linearization" Proceedings of the 2003 American Control Conference, (2003), 3423-3428 vol.4, DOI: 10.1109/ACC.2003.1244061.

[6] M. Albaker Kawtharani, V. Fakhari and M. Reza Haghjoo, "Tracking Control of an Omni-Directional Mobile Robot ", International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), (2020), 1-8, DOI: 10.1109/HORA49412.2020.9152835.

[7] T. N. Khac, N. Thai Huu, M. N. Van and T. Bui Trung, "Adaptive Optimal Control of Four-Wheel Omni Robot using Reinforcement Learning" International Conference on System Science and Engineering (ICSSE), (2021), 164-169, DOI: 10.1109/ICSSE52999.2021.9538431.

[8] J. Cornejo, J. Magallanes, E. Denegri and R. Canahuire, "Trajectory Tracking Control of a Differential Wheeled Mobile Robot: a Polar Coordinates Control and LQR Comparison", IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), (2018), 1-4, DOI: 10.1109/INTERCON.2018.8526366.

[9] A. Sheikhlar, A. Fakharian and A. Adhami-Mirhosseini, "Fuzzy adaptive PI control of omni-directional mobile robot", 13th Iranian Conference on Fuzzy Systems (IFSC), (2013), 1-4, DOI: 10.1109/IFSC.2013.6675667.

[10] K. Kanjanawanishkul, "Path Following and Velocity Optimizing for an Omnidirectional Mobile Robot", International Journal of Engineering-Transaction A: Basics, Vol.28, No. 4, (2015), 537-545.

[11] M.H. Korayem, T.Bani Rostam, "Experimental Analysis for Measuring Errors in Wheeled Mobile Robots", International Journal of Engineering, Vol.18, No. 2, (2005), 115-133.

[12] S.Kim and D.H. Kim, "A design and implementation of moving object tracking system for omni-directional robot," International Conference on Control, Automation and Systems, (2007), 1790-1793, DOI: 10.1109/ICCAS.2007.4406635.

[13] M. Galicki and M. Banaszkiewicz, "Optimal trajectory tracking control of omni-directional mobile robots", 12th International Workshop on Robot Motion and Control (RoMoCo), (2019), 137-142, DOI: 10.1109/RoMoCo.2019.8787377.