

# Autonomous open-source electric wheelchair platform with internet-of-things and proportional-integral-derivative control

Dechrit Maneetham<sup>1</sup>, Padma Nyoman Crisnapati<sup>1</sup>, Yamin Thwe<sup>2</sup>

<sup>1</sup>Department of Mechatronics Engineering, Rajamangala University of Technology Thanyaburi, Pathum Thani, Thailand

<sup>2</sup>Department of Data and Information Science, Rajamangala University of Technology Thanyaburi, Pathum Thani, Thailand

## Article Info

### Article history:

Received Apr 21, 2023

Revised Jul 6, 2023

Accepted Jul 9, 2023

### Keywords:

Electric powered wheelchair  
Internet of things  
Motion control  
Proportional integral derivative  
Pure pursuit  
Unmanned ground wheelchair

## ABSTRACT

This study aims to improve the working model of autonomous wheelchair navigation for disabled patients using the internet of things (IoT). A proportional-integral-derivative (PID) control algorithm is applied to the autonomous wheelchair to control movement based on position coordinates and orientation provided by the global positioning system (GPS) and digital compass sensor. This system is controlled through the IoT system, which can be operated from a web browser. Autonomous wheelchairs are handled using a waypoint algorithm; ESP8266 is used as a microcontroller unit that acts as a bridge for transmitting data obtained by sensors and controlling the direct current (DC) motors as actuators. The proposed system and the autonomous wheelchair performance gave satisfactory results with a longitude and latitude error of 1.1 meters to 4.5 meters. This error is obtained because of the limitations of GPS with the type of Ublox Neo-M8N. As a starting point for further research, a mathematical model of a wheelchair was created, and pure pursuit control algorithm was used to simulate the movement. An open-source autonomous IoT platform for electric wheelchairs has been successfully created; this platform can help nurses and caretakers.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Padma Nyoman Crisnapati

Department of Mechatronics Engineering, Faculty of Technical Education, Rajamangala University of Technology Thanyaburi

39 Village No. 1 Rangsit-Nakhon Nayok Road, Tambon Khlong Hok, Amphoe Thanyaburi Pathum Thani 12110, Thailand

Email: padma\_c@mail.rmutt.ac.th

## 1. INTRODUCTION

This study aims to provide mobility for patients with problems such as spinal cord injuries or neurological problems who cannot use electric powered wheelchairs (EPW) [1], [2]. Intelligence EPWs can move autonomously to the destination point without requiring much human assistance. Intelligence EPW is very helpful for patients with disabilities in carrying out mobility independently [3].

This research is essential because electric wheelchairs are still a daily necessity for patients with disabilities; this is supported by the increasing number of patients, causing hospitals not to provide adequate care to patients [4]. Developing intelligent EPWs is an alternative option that will reduce nurses' workload and help patients in the future. The proposed design of autonomous wheelchairs is to provide integration of the internet of things (IoT) and human machine interface (HMI) to enhance the mobility or movement of autonomous wheelchairs [5]. The wheelchair can be controlled to move in four directions and stop. The HMI system is designed to be easy to use. Several previous studies regarding smart wheelchairs have been carried out by combining electro-mechanical systems, sound sensors, gyro, accelerometer, compass, and

electroencephalogram (EEG) [6]–[13], [14]–[20]. Some previous studies are still in the prototype stage and have not yet entered the testing and implementation stages. Some of these studies have applied them directly to commercial wheelchairs. In this research, wheelchairs designed by the Rajamangala University of Technology Thanyaburi (RMUTT) mechatronics department were used, produced in over a hundred pieces, and donated to several hospitals and communities. Therefore, the autonomous wheelchair was implemented and tested using the waypoint navigation method in this study. This navigation refers to the position values given by the global positioning system (GPS) sensor (longitude, latitude) and the orientation obtained from the compass sensor. This study uses the proportional-integral-derivative (PID) algorithm as a closed-loop control system. The PID algorithm was chosen because it is simple and easy to implement on small microcontroller devices [21].

Field testing has been successfully carried out using three scenarios. The advantages of using this autonomous wheelchair are: i) autonomous wheelchairs can move at a constant speed of not more than 5 km/hour when used correctly, ii) they can reduce the task of nurses or helpers, iii) autonomous wheelchairs can walk by themselves, and iv) can be operated and controlled via mobile phone. In addition, the autonomous wheelchair can gather information and recognize location and displacements through google maps coordinates with longitude and latitude. This research also conducted an initial study on applying the pure pursuit algorithm to wheelchairs using MATLAB/Simulink simulations. This research needs to be done to determine the behavior of wheelchair movements with several parameters specified in the scenarios.

This paper is organized into four sections. In section 2, an outline of the autonomous wheelchair, along with its research methods, including hardware and software development, is presented. Section 3 focuses on the presentation of experimental results and discussion. Finally, in section 4, conclusions for the autonomous wheelchair are provided.

## 2. METHOD

### 2.1. Mechanical design

The wheelchair used in this study is a product of the RMUTT mechatronics department donated to several hospitals and communities, not a commercialized product. This wheelchair is equipped with two DC motors as the prime mover of the rear wheels as a booster. Figure 1 shows the specifications and dimensions of the autonomous wheelchair.

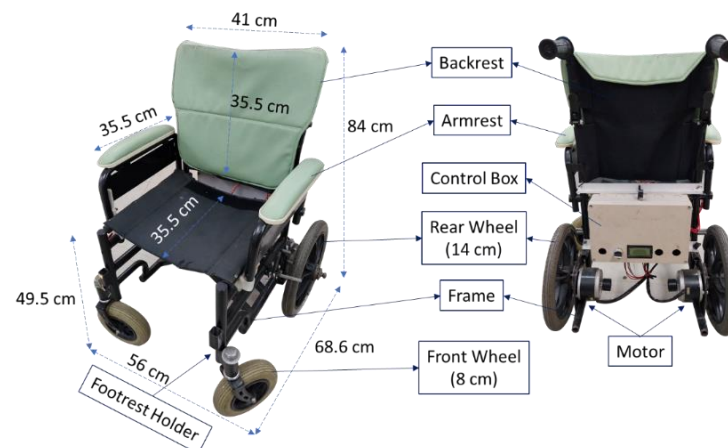


Figure 1. Autonomous wheelchair mechanical design

### 2.2. Controller design

The traditional proportional integral differential (PID) control algorithm is used. PID control is designed to be secure enough to control the autonomous wheelchair; this can be achieved by setting the correct PID parameters: the proportional coefficient  $K_p$ , the integral coefficient  $K_i$ , and the derived coefficient  $K_d$ . PID control is applied to control the position and speed of the autonomous wheelchair [22]. The PID control system block diagram can be seen in Figure 2(a). The hardware connection of this system is shown in Figure 2(b). The proportional term  $P(t_k)$  is used to process gain and measure error. The proportional gain condition can be expressed as (1). The integral term  $I(t_{k+1})$  is properly in the closed loop by eliminating steady-state offset. The integral term can be written as (2). The derivative factor  $D(t_k)$  is

achieved from approximation between the derivative with present error and versus error. The derivative term can be written as (3), where  $v(t_k)$  written in (4), and  $u(t_k)$  provided in (5) [21].

$$P(t_k) = K_p e(t_k) \quad (1)$$

$$I(t_{k+1}) = I(t_k) + K_t T_e(t_k) + K_T T(u(t_k) - v(t_k)) \quad (2)$$

$$D(t_k) = \left(\frac{T_f}{T_f+T}\right) D(t_{k-1}) - \left(\frac{K_d}{T_f+T}\right) (y(t_k) - y(t_{k-1})) \quad (3)$$

$$v(t_k) = P(t_k) + I(t_k) + D(t_k) \quad (4)$$

$$u(t_k) = \text{sat}(v, \text{ulow}, \text{uhigh}) \quad (5)$$

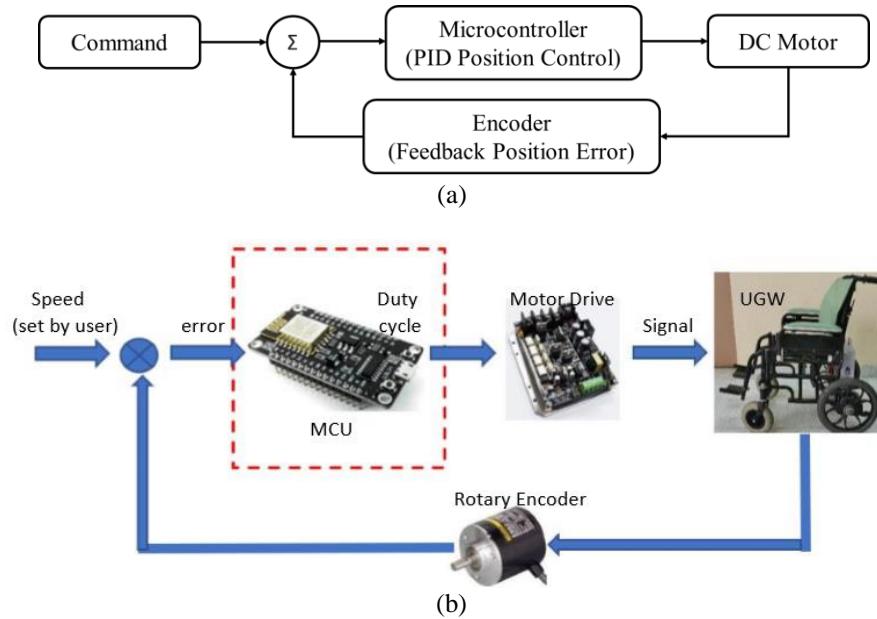


Figure 2. Overview of the (a) the PID closed-loop control system and (b) unmanned ground wheelchair hardware block diagram

### 2.3. Direction and movement

The autonomous wheelchair is controlled according to the direction and the curvature of the desired autonomous wheelchair speed (m/s) and distance (m) of a traditional vehicle. The autonomous wheelchair receives longitude and latitude data from the GPS module (GPS-Ublox M8N). Then, the collected data are compared with the database to generate or define the control direction and position of an autonomous wheelchair for automatic driving. GPS tracking can be used to monitor the route and the actual movement of the autonomous wheelchair. The direction can be obtained as (6), where  $\theta$  is (7) [23].

$$\text{direction} = \tan^{-1} \theta \quad (6)$$

$$\theta = \frac{(x'-x)}{(y'-y)} \quad (7)$$

Wheelchair movement direction is obtained through longitude and latitude coordinates, and the destination point is stored as an array of waypoints. In this case,  $x$  and  $y$  represents the longitude and latitude coordinate functions. In addition, the directional orientation of the autonomous wheelchair is defined by signals from a digital compass (HMC5883L), as shown in Figure 3. The steering wheel (two rear wheels) will turn left or right according to the command from the closed-loop control system. Algorithm 1 shows the pseudocode of the autonomous wheelchair navigation control system.

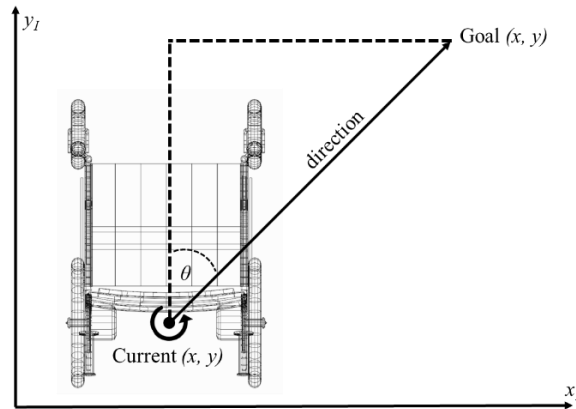


Figure 3. Wheelchair heading system

Algorithm 1. Waypoint navigation system

```

Data: l_goal, c_goal
Result: direction
1. gps_data ← GetGPSData();
2. compass_data ← GetCompassData();
3. while gps_data>0 do
4.   if (l_goal=c_goal) then
5.     direction="break";
6.   else
7.     if (compass_data<=5)
8.       then
9.         direction="move
10.        forward";
11.       else if (CW<CCW) then
12.         direction="turn
13.        right";
14.       else direction="turn
15.        left";
16.     end while
17. return direction;
    
```

2.4. Software design

The internet of things (IoT) has also been designed to access wheelchair navigation controls. This system can be controlled via a cell phone. At the same time, it can monitor and store wheelchair movement data. The cell phone provides communication to control the speed and direction of the wheelchair. In this work, the famous NodeMCU ESP8266 is used. The system will automatically update the database in real-time. This study uses Firebase as a hosting and domain provider to access online HTML, CSS, and JavaScript files. Figure 4 shows the integration of the mechanism, smartphone, primary control, and IoT.

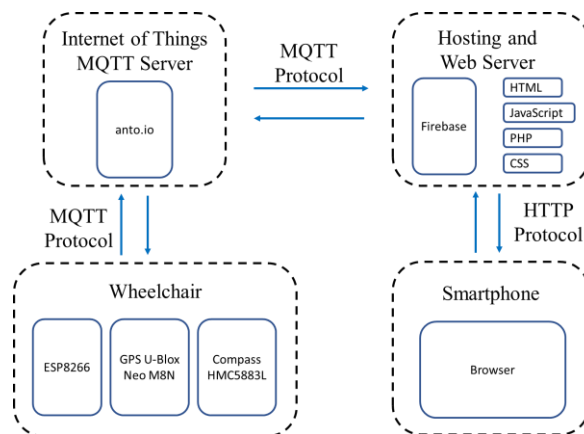


Figure 4. Overview diagram of autonomous wheelchair

**2.5. Simulation design**

The term two-wheel differential drive (TWD) mobile robot model refers to the design and mechanical movement of the robot. Figure 5 describes the coordinate system of the mobile robot to the track line. The wheelchair has two rear wheels as prime movers plus two caster wheels that can rotate at the axis. The robot is limited to linear movement along the x and y axes. The robot’s position in the x and y planes is determined by the movement of the angular velocity of the two driving wheels. Figure 5 illustrates a simplified robot structure. The kinematic model is obtained after conducting field tests and used to simulate the implementation of the pure pursuit control algorithm in the MATLAB application.

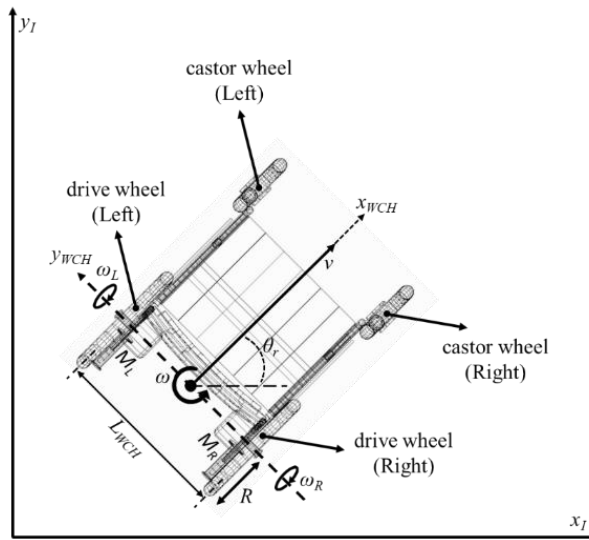


Figure 5. Wheelchair kinematic model

The robot’s pose is supposed to be  $P = (x_r, y_r, \theta_r)$ , where  $x_r$  represents the robot’s x-axis position,  $y_r$  represents the robot’s y-axis position, and  $\theta_r$  indicates the robot’s heading. The robot’s kinematic model can be described as (8) expressing the nonlinear system. Additionally, the robot’s kinematic model can be stated (9). A robot’s linear velocity and the angular velocity open paren(rad/s) describe its mobility. Kinematic equations are calculated using the parameters depicted in Figure 1. The equations (10) and (11) describe the robot’s forward kinematics. The equations (12) and (13) present the robot’s inverse kinematics [24], [25].

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos \theta_r & 0 \\ \sin \theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{8}$$

$$\begin{cases} \dot{x}_r = v \cdot \cos \theta_r \\ \dot{y}_r = v \cdot \sin \theta_r \\ \dot{\theta}_r = \omega \end{cases} \tag{9}$$

$$v = \frac{R}{2} (\omega_R + \omega_L) \tag{10}$$

$$\omega = \frac{R}{L} (\omega_R - \omega_L) \tag{11}$$

$$\omega_L = \frac{1}{R} (v - \frac{\omega L}{2}) \tag{12}$$

$$\omega_R = \frac{1}{R} (v + \frac{\omega L}{2}) \tag{13}$$

$R$  denotes the radius of the rear wheel,  $L$  represents the robot's wheelbase,  $\omega_L$  and  $\omega_R$  denote the left and right wheels' respective angular velocities. By modifying the speed of the two driving wheels, the speed of the wheelchair can be adjusted, as shown in the forward kinematic equation. When the two wheels' speeds are set in opposite directions, the robot can be propelled forward or backward along the x-axis in a straight line. The coordinate system of the wheel is not the same. The linear movement of each wheel shows a different direction of wheel rotation. If the wheelchair moves positively concerning the z-axis, the right wheel must oppose its axis, and vice versa for the left wheel. It must have a negative value on its axis.

**2.6. Pure pursuit control algorithm (PPA)**

The pure pursuit algorithm uses a circular arc as a reference when the wheelchair moves forward toward the specified waypoint. The pure pursuit algorithm (PPA) calculation has a lookahead distance parameter starting from the robot's actual position to the closest point of the reference path. PPA uses a constant linear speed, but the controller produces a different wheel speed value. In Figure 6(a), the radius of the curve connecting the nearest robot location and the reference path  $(x_{obm}, y_{obm})$  to the following reference point  $(x_{ref}, y_{ref})$  is denoted by  $r$ . The variables  $x, y$  show the error between the actual position and the target rather than the wheelchair. In this scenario, the viewing distance,  $L$ , significantly impacts the controller's performance. In (14), (15), (16), (17), and (18) are derived from Figure 6(a) in order to address the issue of determining the turning radius, denoted as  $r$ . By using these equations, the turning radius problem and derive the curvature equation was solved and represented as (19) [26].

$$r = \Delta x + d \tag{14}$$

$$r = d^2 + (\Delta y)^2 \tag{15}$$

$$r^2 = d^2 + (\Delta y)^2 \tag{16}$$

$$r^2 = (r - \Delta x)^2 + (\Delta y)^2 \tag{17}$$

$$r = \frac{(\Delta x)^2 + (\Delta y)^2}{2\Delta x} \tag{18}$$

$$\gamma = \frac{1}{r} = \frac{2\Delta x}{L^2} \tag{19}$$

In the simulation performed on MATLAB/Simulink, the application of PPA calculates the parameter  $v$  as the wheelchair's linear speed and angular velocity with the actual orientation of the wheelchair and the set of waypoints. In Figure 6(b), an illustration of the input and output of PPA is presented.  $I$  is the coordinate frame, while  $x$  and  $y$  are the pose points of the wheelchair. The lookahead value significantly affects system performance; an inaccurate value can cause system instability. Therefore, the lookahead value must be calculated well before field implementation.

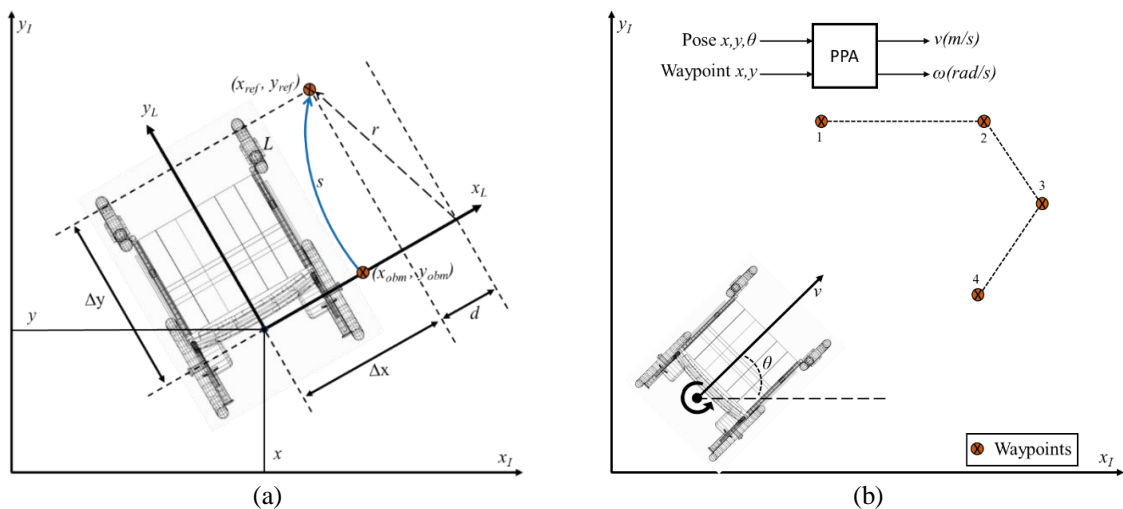


Figure 6. Wheelchair parameters based on (a) PPA strategy and (b) PPA input-output

### 3. RESULTS AND DISCUSSION

Figures 7 to 10 show the results of the autonomous wheelchair experiment in the field. Three field trials were conducted with three types of tracks. Figure 7(a) shows the path following the plane's outline in the counterclockwise direction. Track coordinates can be set on the device according to the map on Google Maps. Coordinates are the points of the location's latitude and longitude. Several dots are arranged in a pattern to form a track. Figure 7(b) shows the system's performance following the path given in Figure 7. The starting point between the assigned route and the actual response is different due to GPS resolution. However, the system is capable of following trajectory commands. The most significant error in latitude between the set value and the actual value at the starting point is about 3 meters. The most significant error in longitude is about 4.5 meters.

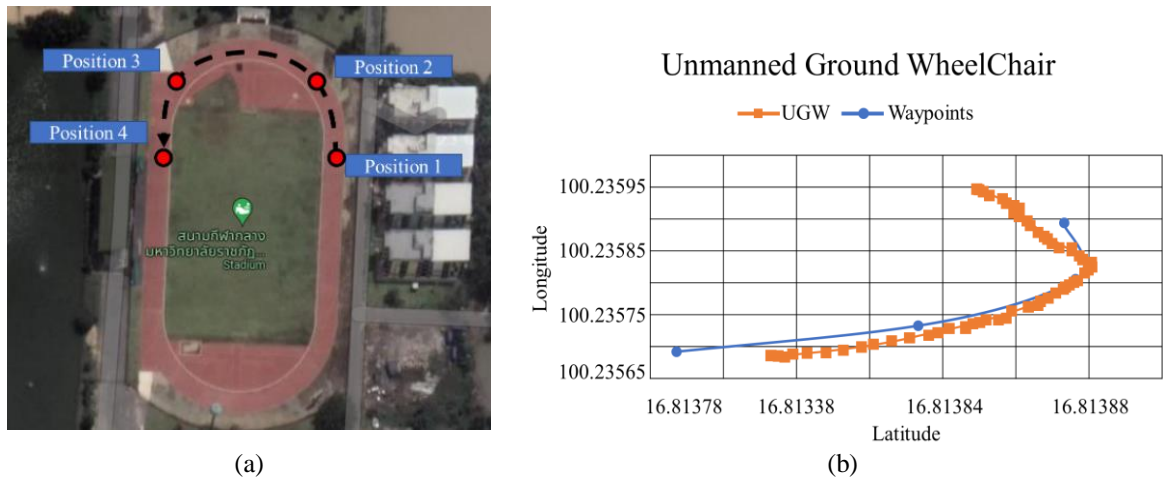


Figure 7. Field trial experiment results (a) curved-up movement scenario and (b) curved movement and setpoint comparison

Figure 8(a) shows the exact path commands as in Figure 7(a) but in a different direction. The system response shown in Figure 8(b) differs from the previous experimental results in Figure 8(a). The second experiment showed that the actual wheelchair starts and end points were close to their set points. But, a moving wheelchair journey from start to finish has a more significant trajectory tracking error than before. The largest error in longitude is about 4.4 meters. The most significant error in latitude is about 1 meter at the start and end points.

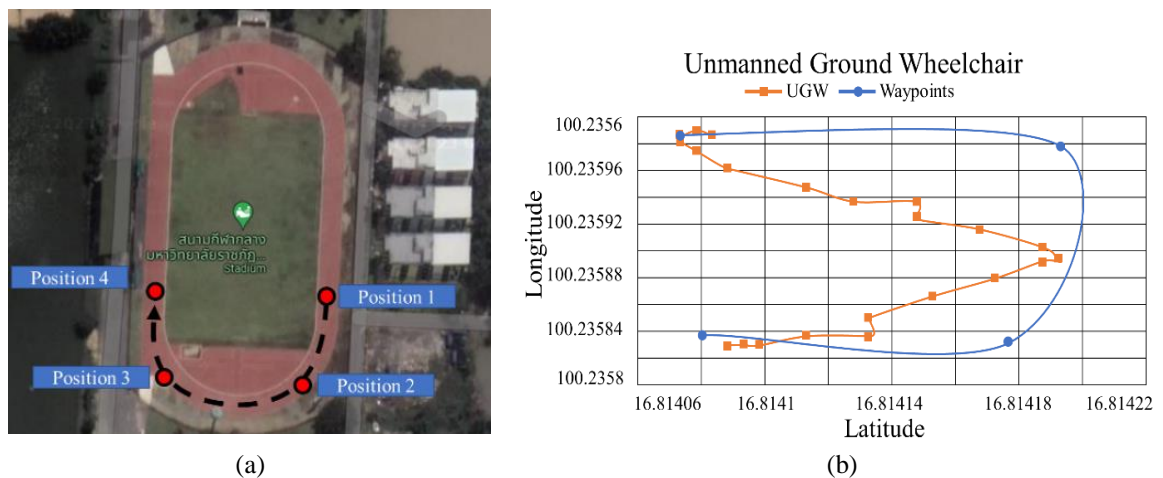


Figure 8. Field trial experiment results (a) curved-down movement scenario and (b) curved movement and setpoint comparison

Figure 9(a) shows a smaller trajectory in a triangular shape. Figure 9(b) shows the experimental results; the wheelchair followed the set point trajectory and made a triangular path. The most significant error in latitude is about 1.1 meters. The largest error in longitude is about 1.8 meters. Figures 10(a) and 10(b) show the results of compass sensor data recording and the actual position of the wheelchair in curved and triangular field trials. Based on the results of field testing, it can be seen that the reading from the compass sensor is very fluctuating; this is due to vibration from the wheelchair and bumpy road conditions.

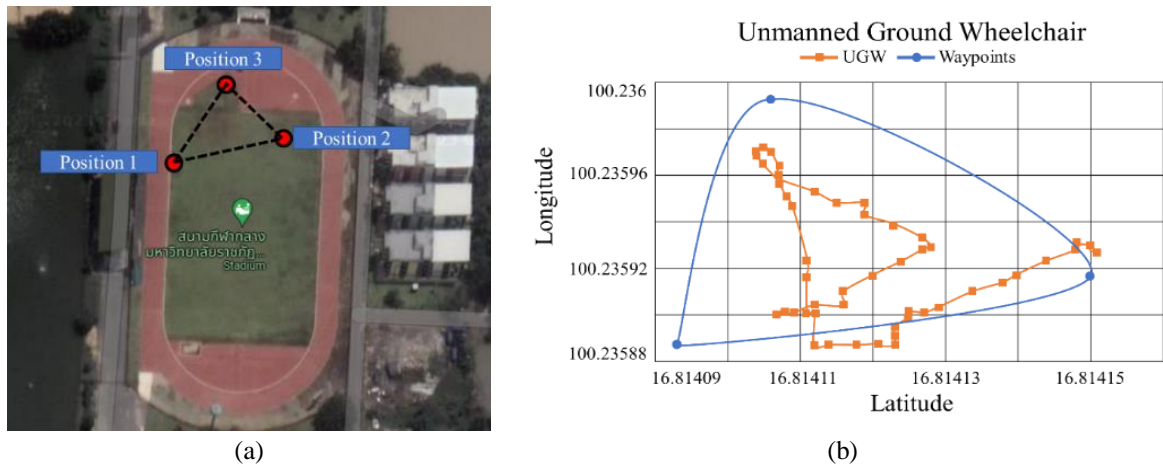


Figure 9. Field trial experiment results (a) triangle movement scenario and (b) movement and setpoint comparison

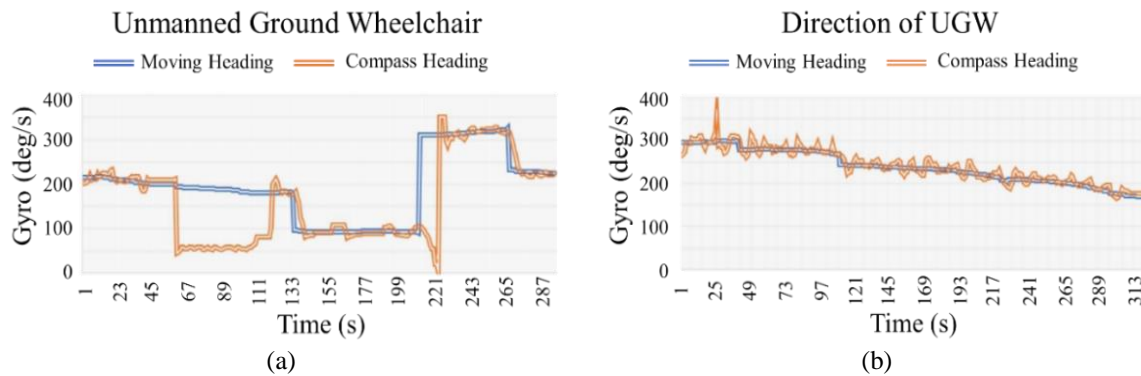


Figure 10. The recorded data from the compass sensor and the wheelchair's actual direction of (a) curved field trial and (b) triangular field trial

To map the running field environment, manually measure the distance between waypoints as shown in Figure 11(a). The measurement results are then reconstructed using Microsoft Paint (MSPaint) to produce a map in .png format. Figure 11(b) shows that one pixel represents one meter; the resulting map resolution is 612×290 pixels. This image is converted into a binary occupancy grid using MATLAB with the 'binaryOccupancyMap' function and saved in matrix format (.mat). The running field has five main waypoints, and the autonomous wheelchair can only move based on the main path that has been determined.

Binary occupancy maps were tested using the probabilistic roadmaps (PRM) tool. PRM allows you to visualize all potential topologies of routes on a map depending on available free space. The PRM algorithm uses a network of connected nodes to find the shortest path between two points to plot paths efficiently in an environment with randomly generated node positions. Figure 12 shows the PRM results. Thus, this map is prepared for the simulation.

Pure pursuit is combined with a range sensor so that the autonomous wheelchair can track its path while avoiding obstacles, as illustrated in Figure 13. The waypoint array is the basis for the autonomous wheelchair moving from one target coordinate to the next. Pose and reach are inputs to determine linear velocity. Next, select the angular momentum of each right and left wheel to get a pose and visualize it. The



detailed input (consisting of pose and coordinates of waypoints) and output (including linear and angular velocity as well as target direction) of the pure pursuit block are presented in Figure 14. These values are utilized alongside the information obtained from the lidar sensor (block vector field histogram) to determine the driving logic of the autonomous wheelchair.

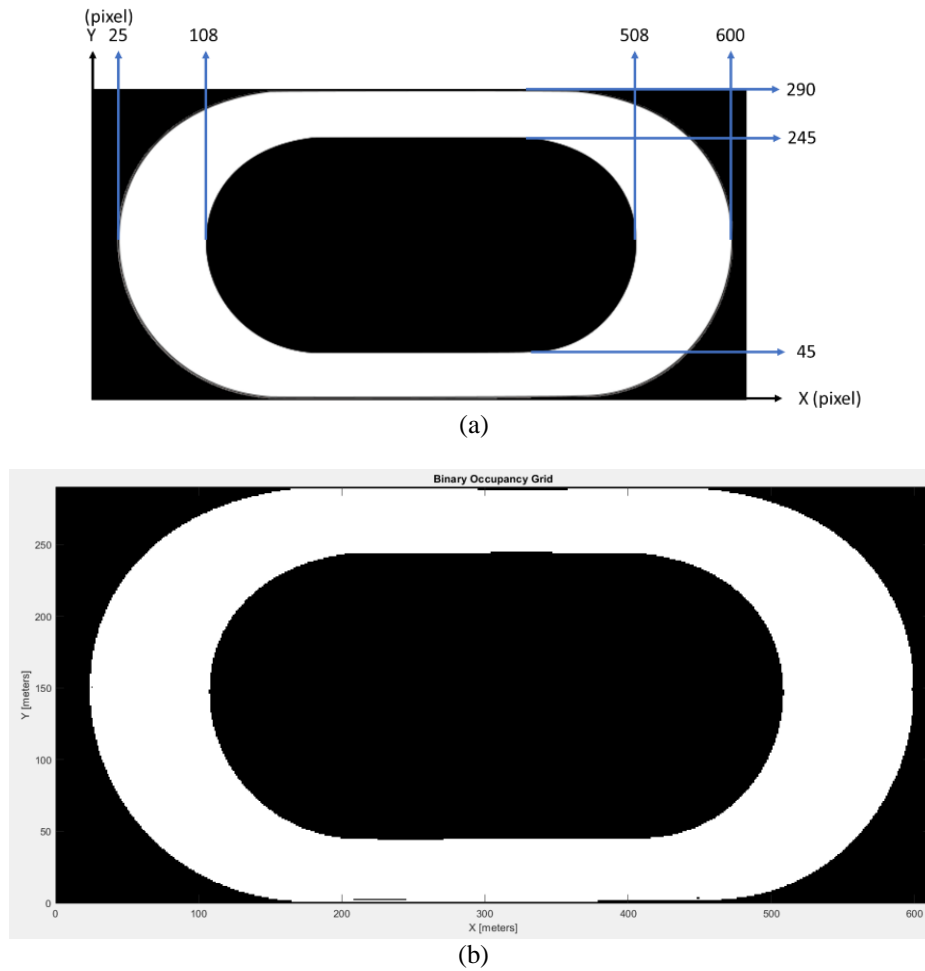


Figure 11. Measurements results (a) wheelchair environment and (b) binary occupancy grid map

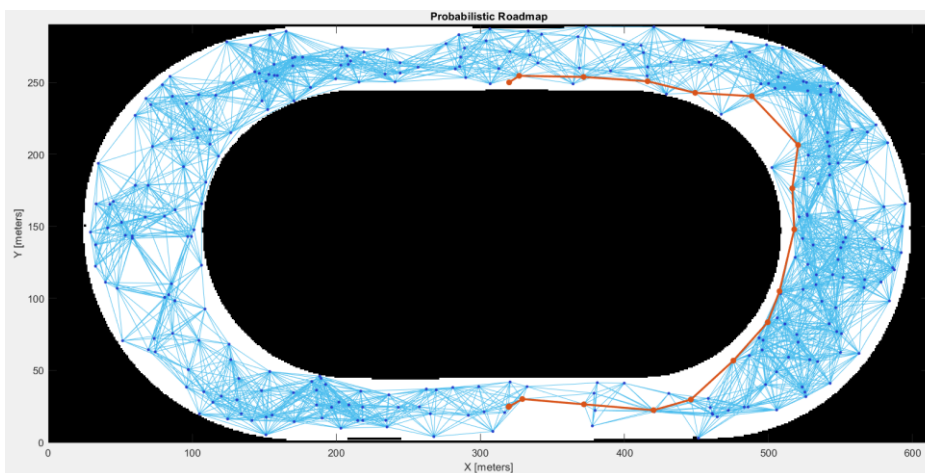


Figure 12. Probabilistic roadmap testing result

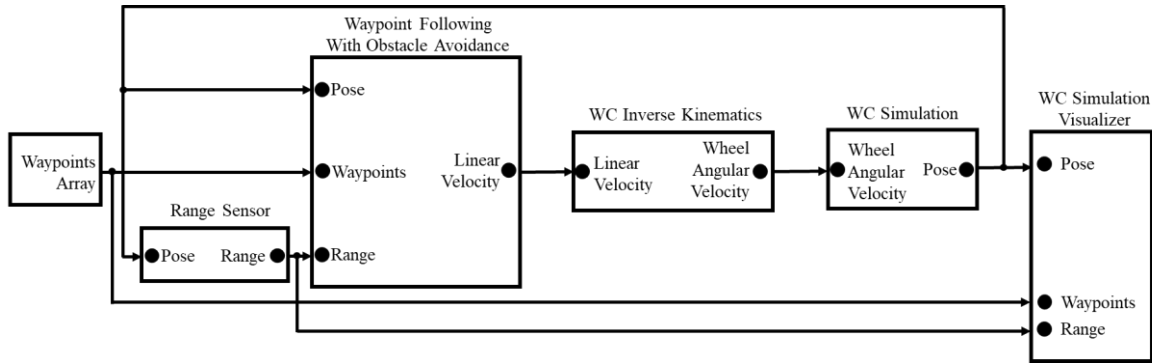


Figure 13. Simulation block diagram

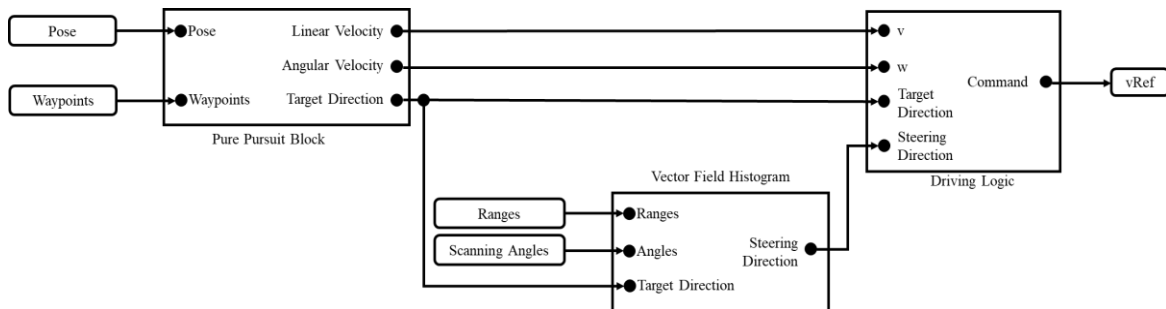


Figure 14. Detail waypoint following with obstacle detection block diagram

Validation was carried out on the kinematics model, binary occupancy map, pure pursuit control, and obstacle detection by providing several test scenarios with different parameters, as shown in Table 1. The simulation scenario of the autonomous wheelchair movement is shown in Figure 15; there are five waypoint coordinates from numbers 1 to 5 with x and y coordinates separated by a semicolon [300 25; 500 25; 525 150; 500 250; 300 250]. The autonomous wheelchair will move from the first waypoint to the next point up to the fifth point.

Table 1. Scenario parameters

Parameters	Scenario			
	S1	S2	S3	S4
Look ahead (m)	0.1	1	4	20
Wheel radius (m)	0.1	0.1	0.1	0.1
Wheelchair radius (m)	0.71	0.71	0.71	0.71
Safety distance (m)	0.2	0.2	0.2	0.2
Min turning radius (m)	0.25	0.25	0.25	0.25
Linear velocity (m/s)	0.75	0.75	0.75	0.75
Angular velocity (rad/s)	1.5	1.5	1.5	1.5
Simulation time (s)	509.7	465.7	466	464.5

The analysis was conducted to determine the optimal lookahead value, and the simulation results are shown in Figures 16(a)-16(b), 17(a)-17(b), 18(a)-18(b), and 19(a)-19(b). This study only considered kinematic equations without accounting for dynamic conditions. A larger lookahead value resulted in smoother tracking. However, with a higher lookahead value, the wheelchair starts turning before reaching the target point. In situations where forward visibility decreases, the robot changes direction after reaching the waypoint, leading to unwanted oscillations in the robot's path. Figures 16(a), 17(a), 18(a), and 19(a) demonstrate that the wheelchair successfully reached the target coordinates. However, upon closer examination of Figure 16(b), there is a noticeable increase in oscillation compared to Figures 17(b), 18(b), and 19(b). In the first scenario with a lookahead value of 0.1 meters, the wheelchair exhibited relatively high oscillation. Therefore, selecting the most suitable value for the lookahead parameter is crucial. Figures 17(b)-19(b) demonstrate relatively stable oscillations in the results, Figure 19(b) successfully

reaches all the specified waypoints with smoother tracking results at a lookahead value of 20 meters. This simulation revealed that a lookahead value of 20 meters is ideal for the autonomous wheelchair to reach all specified waypoints. Figure 20(a) illustrates the speed of the right and left wheels in this simulation, while Figure 20(b) displays the movement of the autonomous wheelchair based on the x and y coordinates.

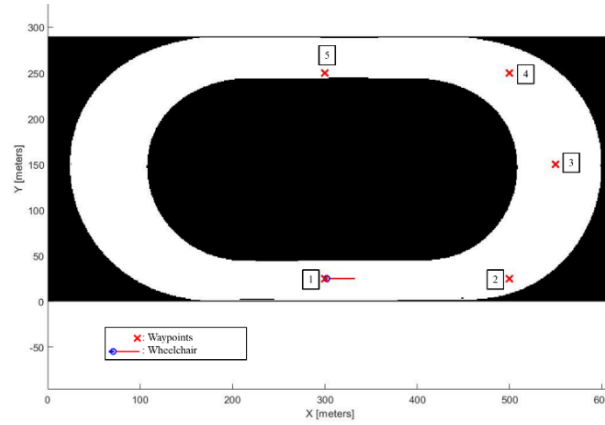


Figure 15. Scenario for autonomous wheelchair simulation

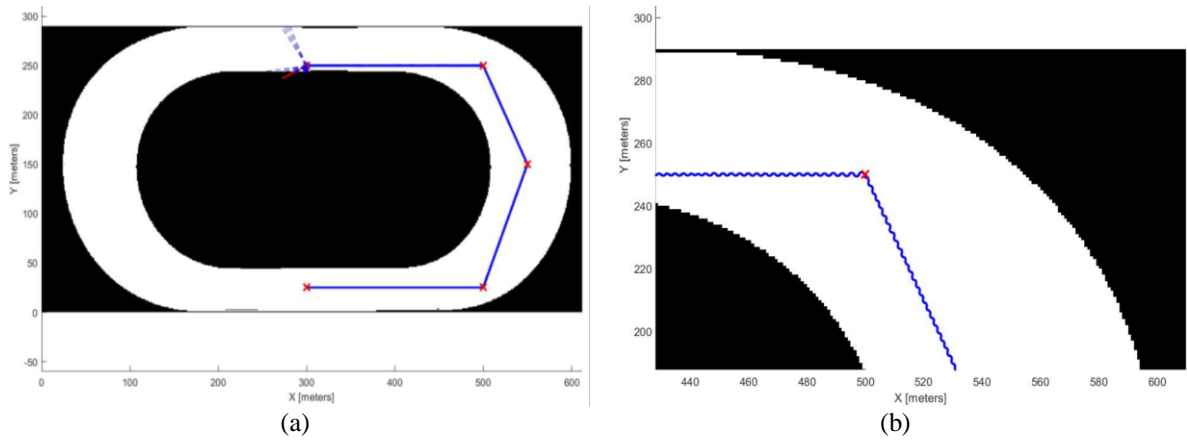


Figure 16. Simulation results based on scenario one (a) S1 and (b) S1 detail

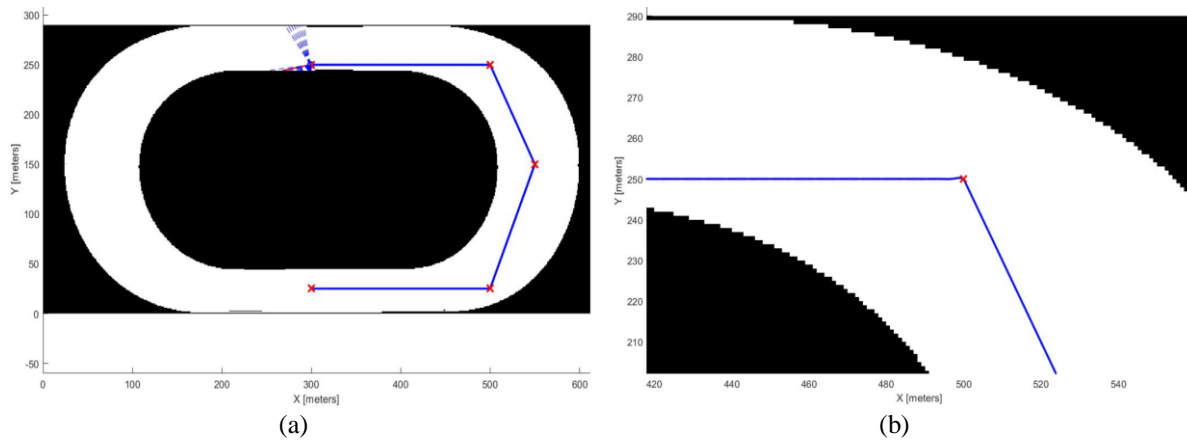


Figure 17. Simulation results based on scenario two (a) S2 and (b) S2 detail

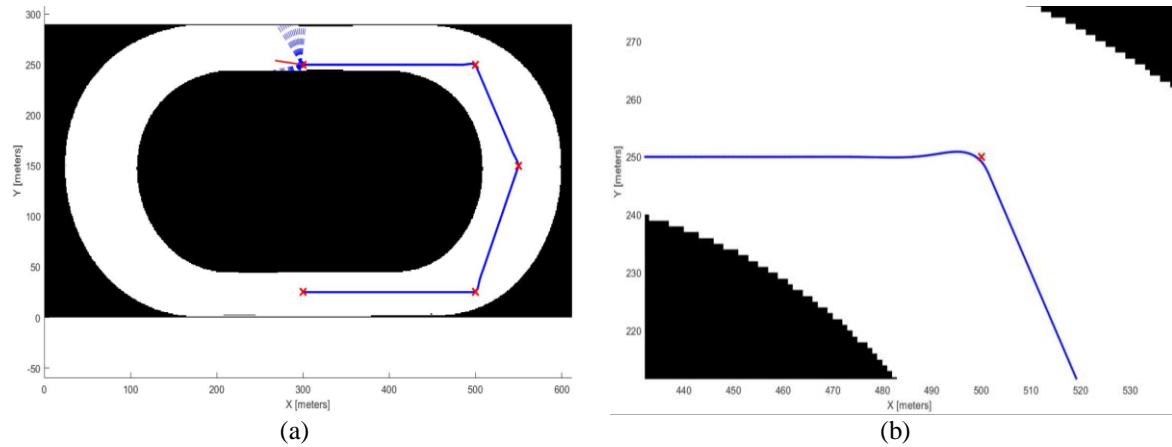


Figure 18. Simulation results based on scenario three (a) S3 and (b) S3 detail

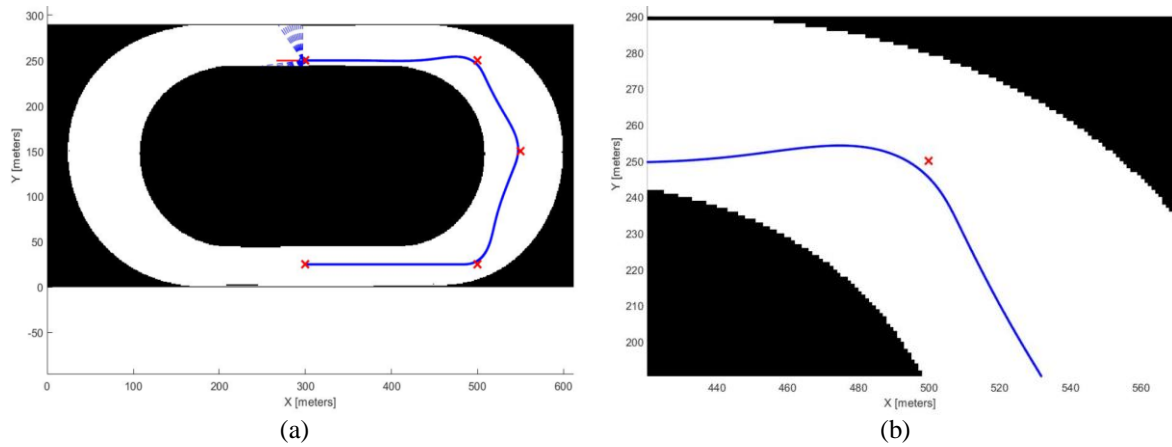


Figure 19. Simulation results based on scenario four (a) S4 and (b) S4 detail

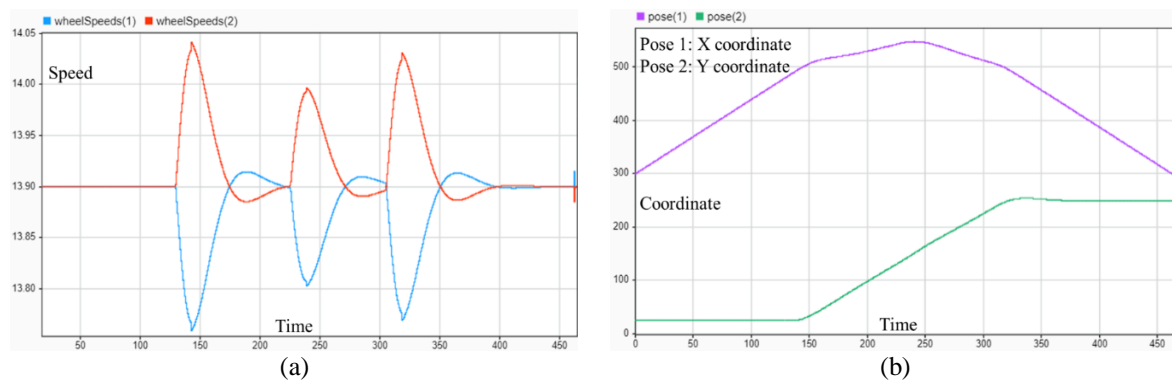


Figure 20. Simulation results based on (a) right and left wheel speeds and (b) X and Y coordinates of the autonomous wheelchair

#### 4. CONCLUSION

The trajectory tracking algorithm has been successfully implemented in the system. The wheelchair can follow the track set point assigned to it. The wheelchair has a more significant error in both latitude and longitude when given a larger track area. When the wheelchair was ordered around the yard, the fundamental error was 4.5 meters. However, when the wheelchair was given a smaller trajectory, the error was also minor. When the wheelchair was ordered to rotate the triangular track route in the yard, the most significant error

was 1.8 meters. Overall, it can be concluded that the autonomous wheelchair can follow a given trajectory setpoint. However, for safety reasons, positional error tolerance must be observed when the wheelchair is used with actual patients. Although the wheelchair can follow given lane commands, it must run properly along the path for the patient's comfort. As such, a kinematic model of the wheelchair was created in this study, serving as the foundation for simulating the wheelchair and implementing the pure pursuit control algorithm. The results of the simulation show that the best value in determining the lookahead distance is 20 meters. Recommendations for further research are the implementation of the pure pursuit algorithm on wheelchairs and increasing heading stability.

## ACKNOWLEDGEMENTS

The appreciation is extended to all those who have provided assistance in terms of human resources, financial support, and facilities during the course of this research. The Mechatronics Engineering Laboratory team at Rajamangala University of Technology Thanyaburi (RMUTT) deserves special acknowledgment for their invaluable support.





## REFERENCES

- [1] L. R. Borges, E. L. M. Naves, and A. A. R. Sa, "Usability evaluation of an electric-powered wheelchair driven by eye tracking," *Universal Access in the Information Society*, vol. 21, no. 4, pp. 1013–1022, Nov. 2022, doi: 10.1007/s10209-021-00809-z.
- [2] F. R. Martins, E. L. M. Naves, Y. Morère, and A. A. R. de Sá, "Preliminary assessment of a multimodal electric-powered wheelchair simulator for training of activities of daily living," *Journal on Multimodal User Interfaces*, vol. 16, no. 2, pp. 193–205, Jun. 2022, doi: 10.1007/s12193-021-00385-9.
- [3] D. Kumar, R. Malhotra, and S. R. Sharma, "Design and construction of a smart wheelchair," *Procedia Computer Science*, vol. 172, pp. 302–307, 2020, doi: 10.1016/j.procs.2020.05.048.
- [4] K. Sasiwongsaroj and Y. Burasit, "Managing Thailand's ageing population," ISEAS Yusof Ishak Institute, 2019.
- [5] J. Cui, L. Cui, Z. Huang, X. Li, and F. Han, "IoT wheelchair control system based on multi-mode sensing and human-machine interaction," *Micromachines*, vol. 13, no. 7, Jul. 2022, doi: 10.3390/mi13071108.
- [6] J. M. Catalan *et al.*, "A modular mobile robotic platform to assist people with different degrees of disability," *Applied Sciences*, vol. 11, no. 15, Aug. 2021, doi: 10.3390/app111517130.
- [7] M. S. Sadi, M. Alotaibi, M. R. Islam, M. S. Islam, T. Alhmiedat, and Z. Bassfar, "Finger-gesture controlled wheelchair with enabling IoT," *Sensors*, vol. 22, no. 22, Nov. 2022, doi: 10.3390/s22228716.
- [8] B. Muangmeesri and K. Wisaeng, "IoT-based discomfort monitoring and a precise point positioning technique system for smart wheelchairs," *Applied System Innovation*, vol. 5, no. 5, Oct. 2022, doi: 10.3390/asi5050103.
- [9] D. Devasia, T. V. Roshini, N. S. Jacob, S. M. Jose, and S. Joseph, "Assistance for quadriplegic with BCI enabled wheelchair and IoT," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, Dec. 2020, pp. 1220–1226, doi: 10.1109/ICISS49785.2020.9315992.
- [10] A. Palumbo, V. Gramigna, B. Calabrese, and N. Ielpo, "Motor-imagery EEG-based BCIs in wheelchair movement and control: a systematic literature review," *Sensors*, vol. 21, no. 18, Sep. 2021, doi: 10.3390/s21186285.
- [11] C. M. Chipaila, D. Grigore, G. Marin, R. Solea, and D. C. Cernega, "Hardware and software solutions for a conventional electric-powered wheelchair," in *2012 16th International Conference on System Theory, Control and Computing (ICSTCC)*, 2012, pp. 1–6.
- [12] E. Yulianto, T. B. Indrato, B. T. M. Nugraha, and S. Suharyati, "Wheelchair for quadriplegic patient with electromyography signal control wireless," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 16, no. 12, Oct. 2020, doi: 10.3991/ijoe.v16i12.15721.
- [13] P. Kanade, J. P. Prasad, and S. Kanade, "IOT based smart healthcare wheelchair for independent elderly," *European Journal of Electrical Engineering and Computer Science*, vol. 5, no. 5, pp. 4–9, Sep. 2021, doi: 10.24018/ejece.2021.5.5.355.
- [14] U. Garg, K. K. Ghanshala, R. C. Joshi, and R. Chauhan, "Design and implementation of smart wheelchair for quadriplegia patients using IOT," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, Dec. 2018, pp. 106–110, doi: 10.1109/ICSCCC.2018.8703354.
- [15] S. A. Akash, A. Menon, A. Gupta, M. W. Wakeel, M. N. Praveen, and P. Meena, "A novel strategy for controlling the movement of a smart wheelchair using internet of things," in *2014 IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS)*, Sep. 2014, pp. 154–158, doi: 10.1109/GHTC-SAS.2014.6967575.
- [16] L. Yang, Y. Ge, W. Li, W. Rao, and W. Shen, "A home mobile healthcare system for wheelchair users," in *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, May 2014, pp. 609–614, doi: 10.1109/CSCWD.2014.6846914.
- [17] S. Kataria, A. S. Menon, P. Sultania, S. S. Paul, and K. A. Kumar, "A novel IOT based smart wheelchair design for cerebral palsy patients," *International Journal of Scientific Research in Science and Technology*, pp. 540–553, Aug. 2021, doi: 10.32628/CEIT2174124.
- [18] S. Soma, N. Patil, F. Salva, and V. Jadhav, "An approach to develop a smart and intelligent wheelchair," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2018, pp. 1–7, doi: 10.1109/ICCCNT.2018.8494050.
- [19] D. A. Sinyukov and T. Padir, "Adaptive motion control for a differentially driven semi-autonomous wheelchair platform," in *2015 International Conference on Advanced Robotics (ICAR)*, Jul. 2015, pp. 288–294, doi: 10.1109/ICAR.2015.7251470.
- [20] A. E. M. Filho and da F. N. J. Viana, "Optimal tuning of dynamic controller via LQR in a powered wheelchair," *American Journal of Engineering Research*, vol. 6, no. 11, pp. 44–53, 2017.
- [21] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control*, vol. 9, no. 2, pp. 818–827, Jun. 2021, doi: 10.1007/s40435-020-00665-4.
- [22] O. A. Somefun, K. Akingbade, and F. Dahunsi, "The dilemma of PID tuning," *Annual Reviews in Control*, vol. 52, pp. 65–74, 2021, doi: 10.1016/j.arcontrol.2021.05.002.





- [23] K. Yeom, "Kinematic and dynamic controller design for autonomous driving of car-like mobile robot," *International Journal of Mechanical Engineering and Robotics Research*, pp. 1058–1064, 2020, doi: 10.18178/ijmerr.9.7.1058-1064.
- [24] A. Filipescu, V. Minzu, B. Dumitrascu, A. Filipescu, and E. Minca, "Trajectory-tracking and discrete-time sliding-mode control of wheeled mobile robots," in *2011 IEEE International Conference on Information and Automation*, Jun. 2011, pp. 27–32, doi: 10.1109/ICINFA.2011.5948958.
- [25] N. Ali, "Fuzzy logic based real time go to goal controller for mobile robot," *International Journal of Computer Applications*, vol. 176, no. 11, pp. 32–36, Apr. 2020, doi: 10.5120/ijca2020920078.
- [26] C. Mahulea, M. Kloetzer, and R. González, *Path planning of cooperative mobile robots using discrete event models*. Wiley, 2019.

## BIOGRAPHIES OF AUTHORS







**Dechrit Maneetham**     is an Associate Professor of Mechatronics Engineering at Rajamangala University of Technology in Thanyaburi, Thailand. He earned his B.Tech. and M.S. in Electrical and Computer Engineering from King Mongkut's University of Technology in North Bangkok, a D.Eng. in Mechatronics from the Asian Institute of Technology in 2010, and a Ph.D. in Electrical and Computer Engineering from Mahasarakham University in 2018. With over 15 years of experience teaching engineering and researching robotics, automation, mechatronics, and biomedical applications, Dr. Dechrit is the author or co-author of more than 30 technical papers and has instructed short robotics courses at various conferences. He has authored seven books, including pneumatics system, hydraulics system, MCS-51 microcontroller, PIC microcontroller, Arduino microcontroller, PLC Beckhoff, and robot. His research interests include biomedical applications, robotics and automation, and mechatronics applications. Dr. Dechrit can be reached via email at: [dechrit\\_m@rmutt.ac.th](mailto:dechrit_m@rmutt.ac.th).



**Padma Nyoman Crisnapati**     after obtaining a Bachelor's degree in 2009 from the Department of Informatics Engineering at Sepuluh Nopember Institute of Technology, Padma pursued a Master's degree in Learning Technology in 2011 and a Master's degree in Computer Science in 2018 from Ganesha Education University. He is a lecturer at STIKOM Bali Institute of Technology and Business, teaching sensors transducer, assembly language, and animation. He previously served as the Head of the Computer Systems Study Program from 2016 to 2020. He is pursuing a Ph.D. in the Mechatronics Engineering department at Rajamangala University of Technology Thanyaburi (RMUTT). His research interests encompass 2D and 3D animation, the internet of things, robotics, automation, augmented and virtual reality, and robotics. For further contact, Padma can be reached via email at: [padma\\_c@mail.rmutt.ac.th](mailto:padma_c@mail.rmutt.ac.th).



**Yamin Thwe**     born in Yangon, Myanmar, in 1997, earned her B.E. degree in Information Technology from the Technological University, Hmawbi, Myanmar, in 2020. As a software engineer for two years, she developed solutions for various E-commerce platforms, government agencies, and non-governmental organizations. Yamin enrolled as an E-CUBE I Scholarship student in the Faculty of Data and Information Science at Rajamangala University of Technology, Thanyaburi, Thailand, in 2021. Her research interests include machine learning, computer vision, and big data security. To contact her, Yamin can be reached via email at: [yamin\\_t@mail.rmutt.ac.th](mailto:yamin_t@mail.rmutt.ac.th).