# AGV maneuverability simulation and design based on pure pursuit algorithm with obstacle avoidance

**Singhanart Ketsayom, Dechrit Maneetham, Padma Nyoman Crisnapati**
Department of Mechatronics Engineering, Faculty of Technical Education, Rajamangala University of Technology Thanyaburi, Thanyaburi, Thailand

**Article Info**

***Article history:***

Received Dec 7, 2023
Revised Jan 22, 2024
Accepted Jan 24, 2024

***Keywords:***

Automated guided vehicle
Control system
Obstacle avoidance
Pure pursuit
Simulation

## ABSTRACT

This paper discusses the simulation of an automated guided vehicle (AGV) with the differential-drive mobile robot (DDMR) concept. Using this wheel configuration, the AGV can maneuver in tight workspaces. However, controlling a self-driving AGV with obstacle avoidance is not easy. Therefore, this paper proposes a control system to drive an AGV with several process stages. First, a kinematic model is formulated to represent the AGV with the concept of two wheels that can be controlled differentially. In the second stage, the pure pursuit control method is applied to the model so that the AGV can follow the waypoint coordinates determined and combine them with obstacle avoidance. Finally, the effectiveness of the control system was verified using simulation. The look-ahead parameter with a value of 0.2 meters shows optimal results so that pure pursuit control can reach all waypoint coordinates. Based on this simulation, the AGV prototype was then designed, assembled, and equipped with an internet of things-based obstacle avoidance system. While the simulation proves promising, the anticipated challenges identified in the AGV field test, such as GPS inaccuracies and signal obstructions, underscore the need for ongoing improvements in real-world applications.

***Corresponding Author:***

Dechrit Maneetham
Department of Mechatronics Engineering, Faculty of Technical Education
Rajamangala University of Technology Thanyaburi
Thanyaburi, Thailand
Email: dechrit_m@rmutt.ac.th

## 1. INTRODUCTION

A warehouse serves as a spacious structure for storing diverse products sourced from various suppliers and arranged on shelves. When a customer makes a purchase, a human picker is required to navigate the warehouse to locate and retrieve the specified product from its designated location, constituting what is referred to as an order-picking operation. These operations incur substantial costs, making up 55% of the overall operating expenses for the warehouse [1]. Managing materials within a warehouse is linked to determining the most efficient route, facilitating optimal transportation for automated navigation vehicles. This aspect is a crucial component in the development of intelligent warehouses.

An autonomous guided vehicle (AGV) is a self-navigating wheeled robot designed to transport shelves autonomously, moving them from storage to a consolidation area [2]. In this consolidation zone, staff retrieve the necessary products from the shelves. Subsequently, the shelves are returned to the storage area using a warehouse mobile robot. The primary function of the warehouse mobile robot is to streamline order fulfilment operations by handling material transport within the warehouse. AGVs typically come equipped with automatic navigation systems, such as electromagnetic, optical, or laser devices [3]. These technologies ensure that the system can autonomously follow a pre-established route without the need for manual steering.

The warehouse moving robot relies on a path-planning algorithm to move materials in the warehouse. Path planning involves constructing a starting point to a destination where the robot must travel. The AGV must run along the built track; the track must be such that the way does not collide with the racks in the warehouse [4]. AGV movement relies on a combination of sensor-based guidance and a software-based guidance system. They have a predictable path for their movement, including obstacle avoidance and detection [5]. This includes transporting goods and raw materials around industries or warehouses to avoid obstacles and ensure safe delivery at their destination. Material handling with the help of AGV is also essential in facilities such as warehouses, factories, terminals, and distribution centers. Autonomous navigation uses a control structure architecture based on automated guided systems in terms of algorithm design.

Based on previous research, there are several categories of groups for mobile robot control systems. One of the categories is the use of sensors in the control system. The use of sensors allows the control system to move the robot in a dynamic environment [6]. Shamsuddin *et al.* [7], research was conducted on controls that can estimate and predict a vehicle's direction based on data obtained by sensors. The use of sensors is also combined with an intelligent control system [8]–[10]. The limited movement of the robot, which is only around track detection, is an obstacle for this category [11]. The next category uses a path planning approach where the control system moves the robot based on a predetermined starting point, path, and endpoint [12], [13]. In addition, mapping is carried out on the surrounding environment so that the robot is avoided from collisions. This category utilizes control strategies that consider distance, time, and energy efficiency. The third category is a strategy that relies on optimization techniques designed to generate proper trajectory control for the robot. The mathematical model of mobile robots forms the basis of this category of controllers. Measurement of the error between the reference and the actual path is also carried out in this category [14], [15]. One of the control methods in this third category is pure pursuit [16], [17].

Based on several previous studies that have successfully implemented several categories of mobile robot systems with sensors that are classified as high-cost, this paper proposes to simulate, design, and implement an AGV using a combination of three categories combining pure pursuit control, waypoints, and several low-cost sensors (GPS, range, and magnetometer) to reach the desired target. Using this scheme, the AGV has better waypoint tracking capabilities and shorter path lengths. In this paper's case study, a binary occupancy grid map is employed to represent the storage environment, optimize path planning, address obstacle avoidance, and validate the algorithm's efficacy through simulations conducted in MATLAB/Simulink. In contrast to previous research, which only reached the prototype and simulation stages, this research carried out direct implementation and testing in the field. The map was made based on real situations on the shop floor of K. Rattanakit Co., Ltd., Thailand. The company trades or manufactures building materials, chemical products and operates road services. The AGV simulated approach has been successfully tested on the actual factory floor using several inexpensive sensors, including the HMC5883l Magnetometer and the Ublox-6M GPS. Field testing revealed that the AGV successfully adhered to several preset points. Several updates still need to be made to refine AGV movements in tracing the waypoints that have been given. Details of the research results are presented in the following sections.

## 2. METHOD

In general, as shown in Figure 1, this research methodology starts from designing the AGV regarding the length, width, and height of the AGV, distance between wheels and wheel diameter. Based on this design, a kinematics model is then prepared, which will be used as a basis for parameters in creating simulations in the MATLAB application. At the simulation stage, several tests were carried out using several scenarios to obtain the most optimal look-ahead value. After that, the system is then implemented in actual conditions on the factory floor to distribute material from one point to another.

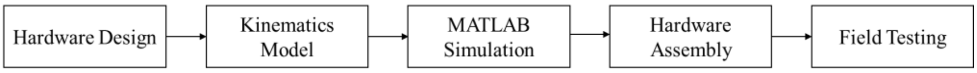


Figure 1. Research methodology

### 2.1. Warehouse AGV environment modeling

Manual measurements are taken between waypoints to create a representation of the warehouse environment, as illustrated in Figure 2. The distances measured are then employed in an image editing application to generate a map saved in .png format, as illustrated in Figure 3. It is noteworthy that each pixel in the resulting map represents one meter, resulting in a resolution of 37×22 pixels. Subsequently, using the applications and libraries provided by MATLAB, images can be converted into a binary occupancy grid, and they are saved in matrix (.mat) format. The warehouse comprises four primary locations: pick-up point, warehouse one, warehouse two, and charging point. AGV movement is constrained to predefined main roads, with the AGV navigating based on waypoints positioned in front of each room, sequentially from room number 1 to 4. In the simulation, the waypoints are represented by x and y coordinates on the Cartesian axis. However, in the actual implementation, these coordinates will be replaced with longitude and latitude values. The obstacles avoided in the simulation are walls that divide the robot's movement.

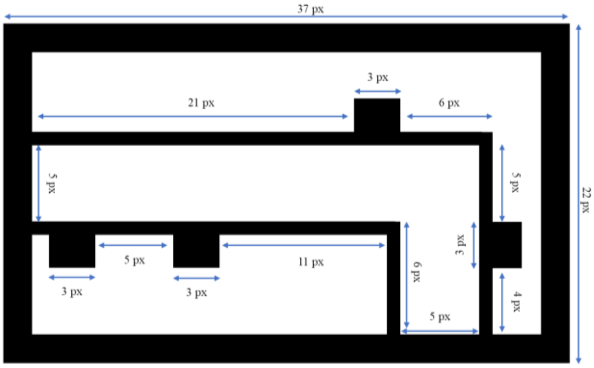


Figure 2. AGV environmental measurement

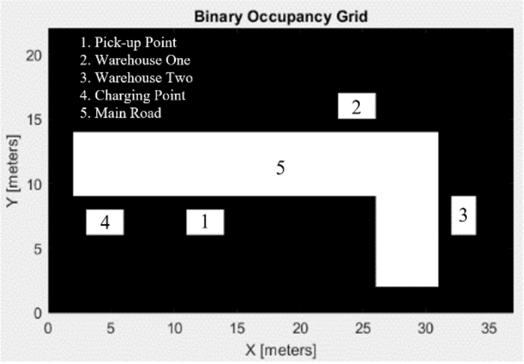


Figure 3. Binary occupancy grid map

## 2.2. AGV hardware design and kinematics model

The model of the mobile robot, known as the two-wheel differential drive (TWD) [18], [19], refers to both the mechanical structure of the robot and its method of movement. Figure 4 depicts the Cartesian coordinate system of the AGV with the waypoint trajectory. The robot has two rear wheels powered by a motor, while the third wheel is a caster that can rotate on its axis. This configuration allows the robot to move exclusively along the *x*-axis and precludes movement along the *y*-axis. Moreover, the robot possesses the capability to rotate on the *z*-axis. Modifying the angular speed differential between the left and right wheels allows the robot to move in the *x* and *y* planes.

The robot's chosen pose is represented as $Pose = (x_r, y_r, \theta_r)$, where $y_r$ represents the robot's *y*-axis location, $\theta_r$ its direction, and $x_r$ its position along the *x*-axis. The (1) illustrates the AGV's kinematic model, which is further expressed in (2). *v* in units of m/s is a linear velocity, and $\omega$ in rad/s is angular velocity. Some of the parameters used in this kinematic equation are illustrated in Figure 4. The forward kinematics of the robot are provided in (3) and (4), while the inverse kinematics are presented in (5) and (6) [20], [21].

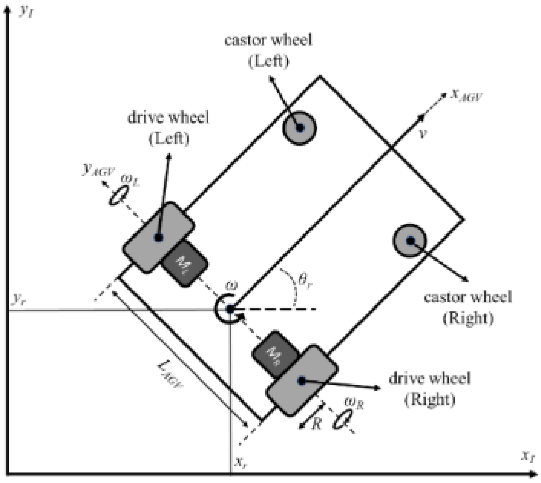


Figure 4. The kinematic model of the AGV

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

$$\begin{cases} \dot{x}_r = v.\cos\theta_r \\ \dot{y}_r = v.\sin\theta_r \\ \quad \dot{\theta}_r = \omega \end{cases} \tag{2}$$

$$v = \frac{R}{2}(\omega_R + \omega_L) \tag{3}$$

$$\omega = \frac{R}{L}(\omega_R - \omega_L) \tag{4}$$

$$\omega_L = \frac{1}{R}\left(v - \frac{\omega_L}{2}\right) \tag{5}$$

$$\omega_R = \frac{1}{R}\left(v - \frac{\omega_L}{2}\right) \tag{6}$$

In the context of the given equations, where $R$ represents the radius of the left/right wheel, $L$ is the wheelbase of the robot, $\omega_L$ and $\omega_R$ are the angular velocities of the left and right wheels, respectively; the manipulation of the left and right wheel speeds becomes the means by which the robot is controlled. By using the forward kinematics equation, the speed between the two AGV wheels can move in opposite directions so that linearly along the $x$-axis, it can navigate forward or backwards. Notably, the wheel coordinate system lacks uniformity, requiring distinct wheel rotation directions for linear motion. Suppose the robot rotates positively about the $z$-axis. In that case, it is crucial to assign a positive orientation to the left wheel along its axis and a negative orientation to the right wheel along its axis.

### 2.3. AGV control algorithm

The underlying principle of pure pursuit's control laws involves iteratively fitting distinct circular arcs to various waypoints as the robot progresses forward, ultimately reaching the final destination point [14]. The destination is defined as a constant look-ahead distance along the path, measured from the closest point between the AGV's current location and the reference path. The output of the controller establishes the wheel speed, ensuring a consistent and steady linear velocity for the AGV. Figure 5 presents an illustrative depiction and explanatory breakdown of the Pure Pursuit Algorithm (PPA), a widely used method in robotics and autonomous vehicle navigation. Figure 5(a) shows that the symbol "$r$" signifies the curvature radius, indicating the distance from the closest robot position and reference path to the subsequent reference point. The variables $x$ and $y$ denote the longitudinal and lateral errors between the robot's current and intended positions. In this scenario, the specification of the viewing distance, denoted as "$L$" significantly influences the controller's performance. Utilizing information from Figure 5(a), (7), (8), (9), (10), and (11) are derived to address the turning radius challenge ($r$), ultimately leading to the formulation of the curvature equation in (12) [22], [23].

$$r = \Delta x + d \tag{7}$$

$$r = d^2 + (\Delta y)^2 \tag{8}$$

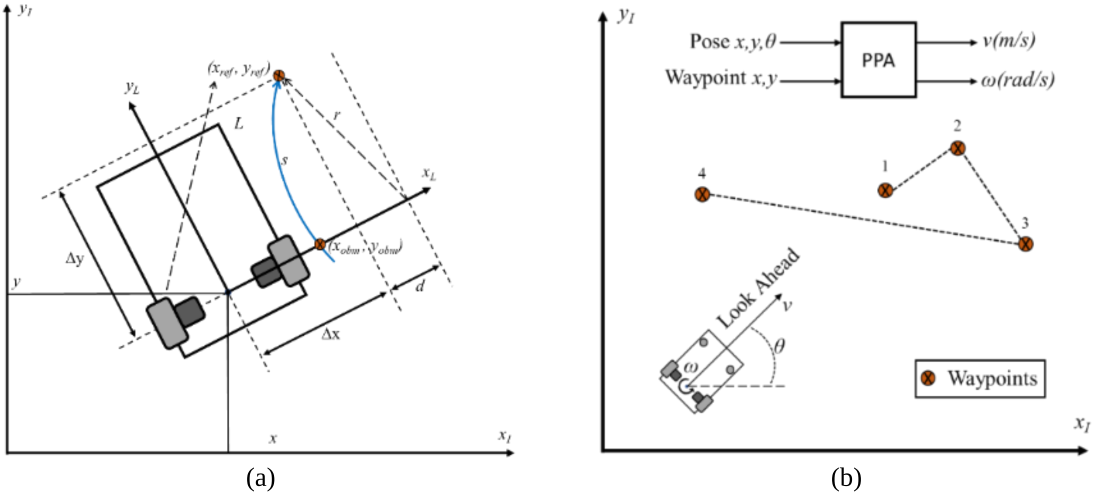
$$r^2 = d^2 + (\Delta y)^2 \tag{9}$$

$$r^2 = (r - \Delta x)^2 + (\Delta y)^2 \tag{10}$$

$$r = \frac{(\Delta x)^2 + (\Delta y)^2}{2\Delta x} \tag{11}$$

$$\gamma = \frac{1}{r} = \frac{2\Delta x}{L^2} \tag{12}$$

The implementation of pure pursuit algorithm (PPA) in the MATLAB/Simulink application computes the linear velocity and angular velocity of the robot based on its current pose and a set of waypoints [24]. The PPA involves two inputs and two outputs, as depicted in Figure 5(b). The pose encompasses the robot's position in the $xy$ plane, with its $x$ and $y$ coordinates relative to the I coordinate frame and their angles relative to the *x*-axis. The forward visibility influences the algorithm's performance. Opting for a smaller visibility value can enhance the robot's path tracking, although an incorrect setting of foresight may lead to system instability. Therefore, it is crucial to determine this value with precision.



(a) (b)

Figure 5. Pure pursuit algorithm (PPA) (a) PPA overview and (b) PPA ways of working

## 3. RESULTS AND DISCUSSION

Based on the research methodology explained previously, in this section, the results and discussions of the experiments that have been carried out will begin to be discussed. This section starts from preparing maps for simulation to creating block diagrams with parameters adapted to the AGV design, which have been discussed in the kinematics model and control algorithm sections. After obtaining the best parameter results in the simulation, system implementation and field testing are then carried out. The results of this research are presented in the following subsections.

### 3.1. Probabilistic roadmap

The probabilistic roadmap function is used to test the binary occupancy map [25], [26]. By employing a probabilistic roadmap, it becomes possible to observe various potential network graphs on the map, considering unoccupied areas. The algorithm for probabilistic roadmaps uses a network of interconnected nodes to detect paths without obstacles, ranging from the initial point to the destination. It efficiently devises routes through the environment by randomly generating the locations of these nodes. Figure 6 is the result of PRM, obtained four-node coordinates with *x* and *y* values separated by the following semicolon [5 10; 5.39 9.72; 30.58 9.43; 30 8]. Thus, this map is ready to be used for simulation.

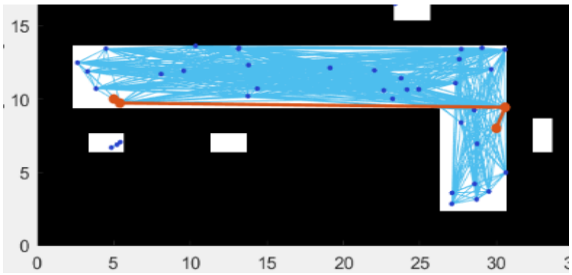


Figure 6. Probabilistic roadmap testing result

### 3.2. MATLAB/Simulink block diagram

By combining range sensors with pure pursuit, the AGV achieves obstacle avoidance while navigating its path, as illustrated in Figure 7. Utilizing a series of waypoints, the AGV advances from one target coordinate to the next. The inputs for determining linear velocity involve pose and range information.

Subsequently, angular momentum for both the right and left wheels is selected to derive the pose and visualize the trajectory. The input and output specifics of the pure pursuit block, integrated with sensor-derived values that determine the AGV's driving logic, are elucidated in Figure 8.

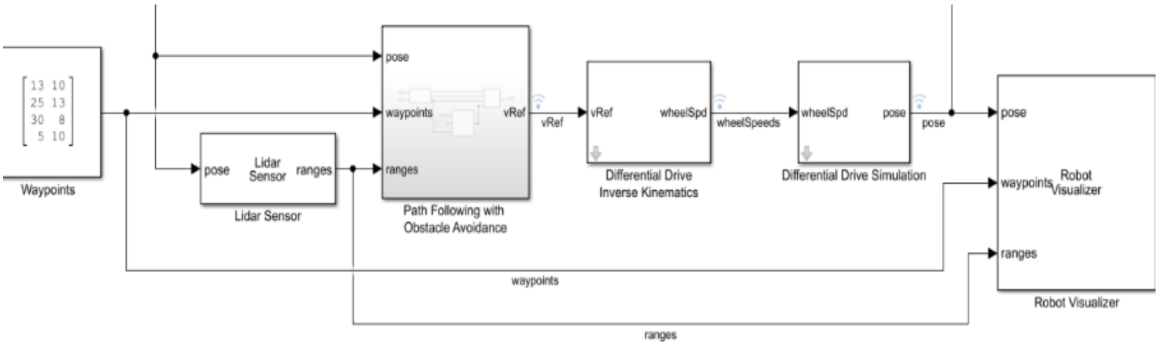


Figure 7. Simulation block diagram

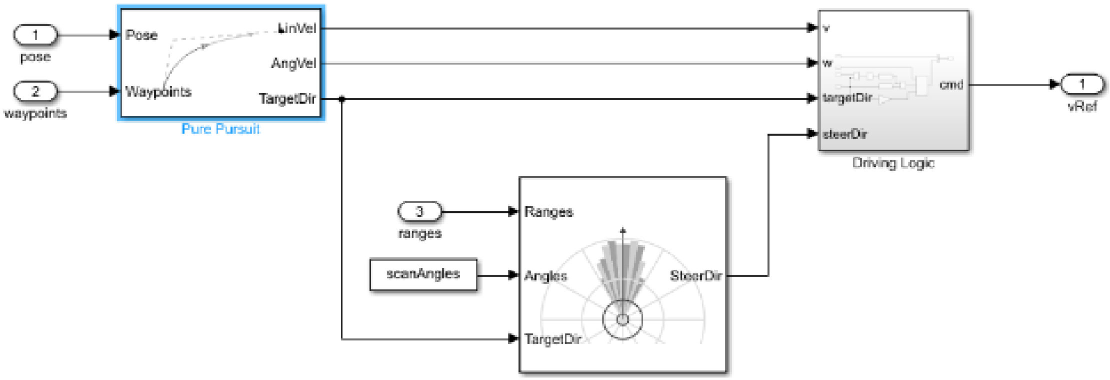


Figure 8. Block diagram of waypoint navigation system and obstacle avoidance

Additionally, Figure 9 provides insights into the inverse kinematics block of the AGV. The angular velocity generated from the previous block is used as input for the AGV simulation block, consisting of AGV forward kinematics and generic robot simulation blocks as shown in Figure 10. Figure 11 shows the details of the forward kinematics block, while Figure 12 is an AGV kinematics block model. To guarantee the precision of the developed kinematic model, pure pursuit control, binary occupancy map, and obstacle detection, it is essential to conduct testing activities that encompass a variety of parameters, as specified in Table 1.

The simulated movement task for the AGV is illustrated in Figure 13, showcasing four waypoint coordinates labelled from 1 to 4. Each coordinate is defined by *x* and *y* values, presented as pairs separated by semicolons [13 10; 25 13; 30 8; 5 10]. The AGV will traverse from the initial waypoint to subsequent points, progressing through the sequence until reaching the fourth point.

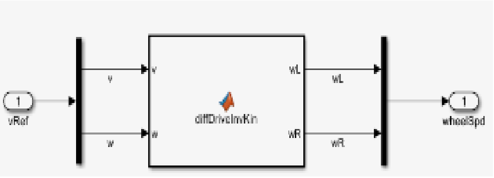


Figure 9. Differential drive inverse kinematics

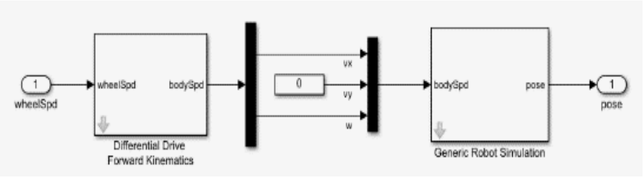


Figure 10. AGV simulation

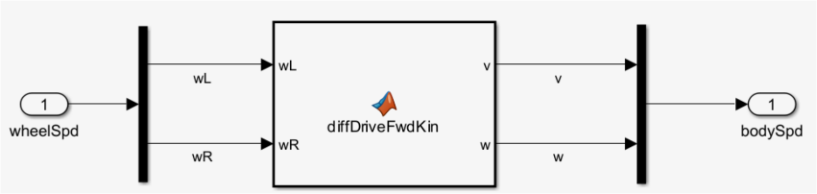


Figure 11. Differential drive forward kinematics

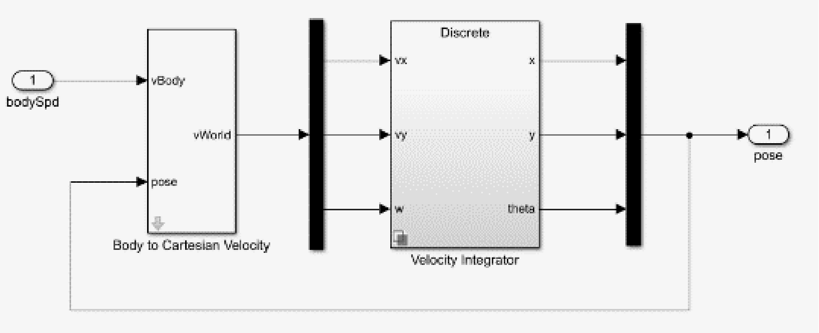


Figure 12. AGV kinematic model block diagram

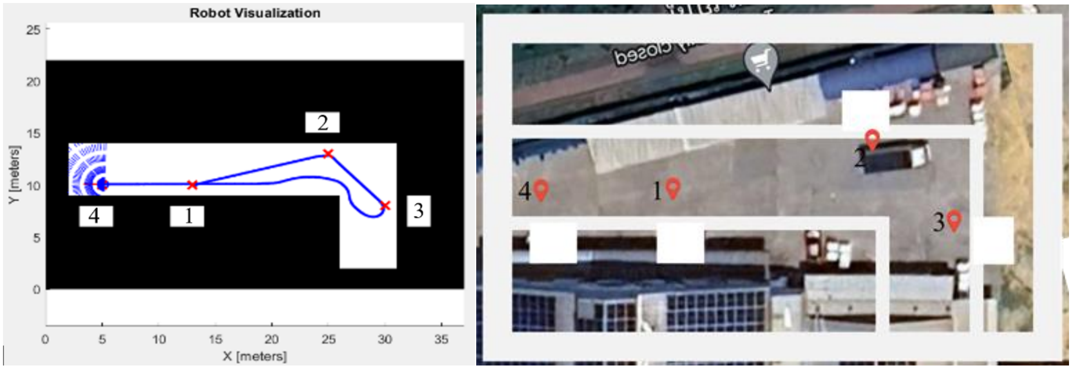


Figure 13. The task for AGV simulation

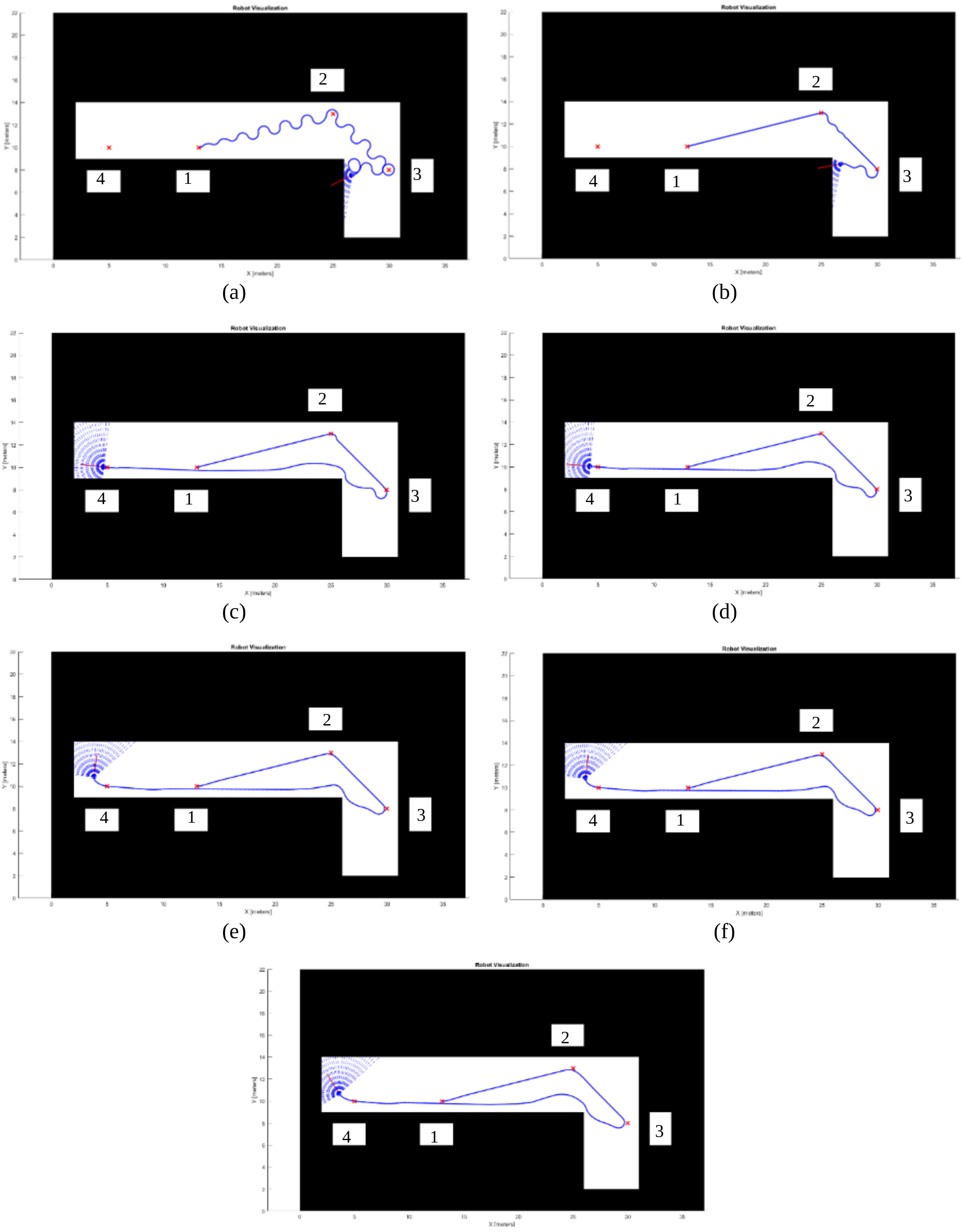
Table 1. Experiment setup

| Parameters | Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Task 6 | Task 7 |
|---|---|---|---|---|---|---|---|
| Look ahead (m) | 0.01 | 0.1 | 0.2 | 0.4 | 0.6 | 0.8 | 1 |
| Wheel radius (m) | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| AGV radius (m) | 0.71 | 0.71 | 0.71 | 0.71 | 0.71 | 0.71 | 0.71 |
| Safety distance (m) | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Turning radius (m) | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 |
| $v$ (m/s) | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 |
| $\omega$ (rad/s) | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| Time (s) | 62.5 | 62.5 | 62.5 | 62.5 | 62.5 | 62.5 | 62.5 |

### 3.3. Experimental results

Figure 14 illustrates the analysis of the look-ahead effect across various values. A smoother path tracking is achievable with an increased look-ahead value, albeit with a directional change occurring before reaching the waypoint. When forward visibility diminishes, the robot alters its course upon reaching the waypoint, resulting in undesirable oscillations, as depicted in Figures 14(a) and 14(b). This study exclusively employs kinematic equations, neglecting dynamic conditions. Consequently, the path taken by the robot exhibits heightened oscillations and covers a greater distance, evident in the blue line when the foresight value is set to 0.01 m or 0.1 m.

Figures 14(c) to 14(g) portray results with relatively stable oscillations, but only Figure 14c successfully reaches all specified waypoint coordinates with a look-ahead value of 0.2 m. The simulation determines that the optimal condition for the AGV to reach all designated waypoints is a look-ahead value of 0.2 m. Figure 15 displays the left and right wheels' simulated speeds, while Figure 16 shows the AGV's movement in relation to its *x* and *y* coordinates.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 14. Experimental results: (a) Task 1; (b) Task 2; (c) Task 3; (d) Task 4; (e) Task 5; (f) Task 6; and (g) Task 7

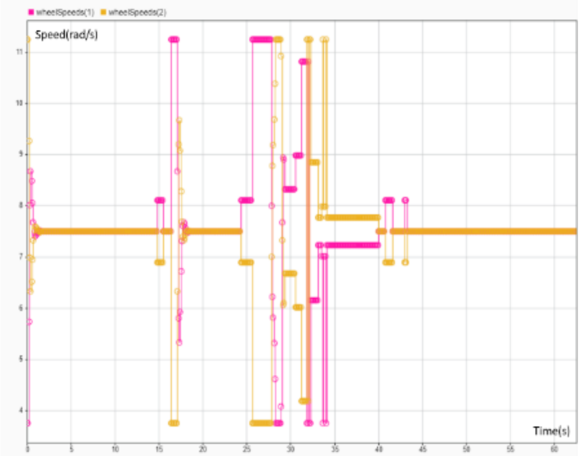


Figure 15. Right and left wheel speeds

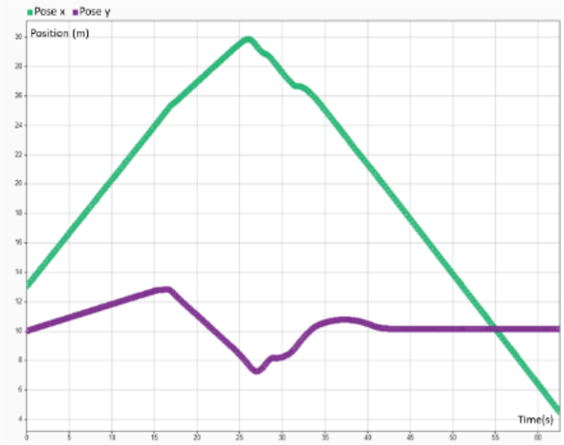


Figure 16. *x* and *y* coordinates of AGV

### 3.4. AGV hardware assembly and test drive

After the simulation, the design and implementation of the AGV from a mechanical and electrical perspective are then carried out. AGV radius and wheel radius are made the same as the parameters used in the simulation. Figure 17 showcases the design and implementation process of an Automated Guided Vehicle (AGV), providing a comprehensive overview through a 3D CAD model, top view, bottom view, and the final assembly configuration. Figures 17(a) to 17(d) show the results of the mechanical design and implementation, while Figure 18 shows the electronic configuration of the AGV robot. Table 2 shows the specifications of the AGV robot obtained through testing in Figure 19.

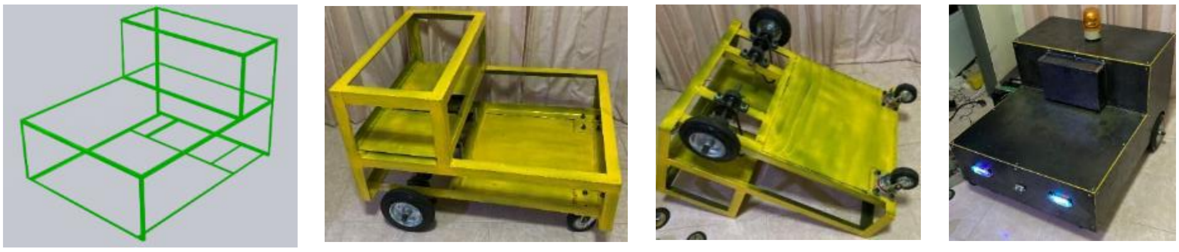

(a) (b) (c) (d)

Figure 17. AGV design and implementation: (a) 3D CAD model, (b) top view, (c) bottom view, and (d) AGV final assembly

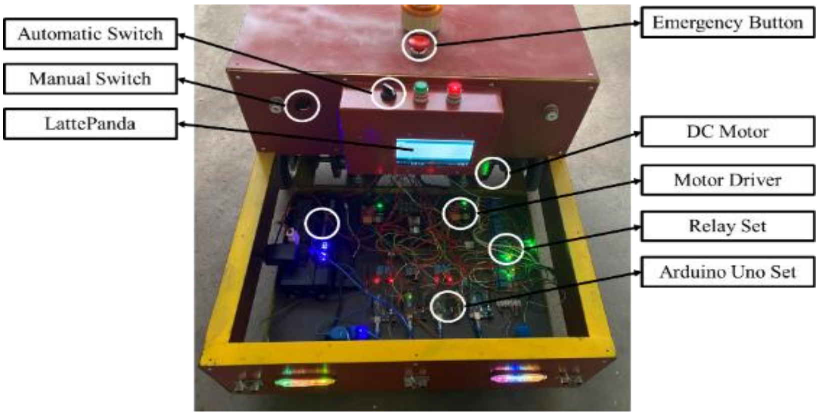


Figure 18. AGV electrical configuration

Figure 19. AGV driving test

Table 2. AGV specification

| Specification | Value | Specification | Value |
|---|---|---|---|
| AGV width | 71 cm | Back wheel radius | 10 cm |
| AGV height | 87 cm | Weight of the AGV | 7.5 kg |
| AGV depth | 58 cm | AGV load capacity | 60-120 kg |
| Distance between rear wheels | 50 cm | DC motor speed | 150-300 RPM |
| Distance between front wheels | 70 cm | DC motor torque | 5-15 Nm. |
| Distance between the front and rear wheel | 60 cm | DC motor voltage | 24 V |
| Front-wheel Radius | 5 cm | Maximum speed | 10-15 Km/Hr. |

### 3.5. Ultrasonic distance scanner (UDS) using the internet of things (IoT)

In addition to the hardware assembly system, a prototype ultrasonic distance sensor is also designed and manufactured using internet of things (IoT) technology. IoT is used with the consideration that this technology allows real-time monitoring of ultrasonic distance scanner (UDS) [27]. This prototype uses a 7.4 V 2,200 mAh Li-Po battery as a power source plugged into a three-legged on/off switch. The switch output is connected to the DC pin input of the L289N motor driver. The only part of the L298N that will be used is the 5 V regulator to reduce the voltage from the battery. Then, the ground and 5 V from L298N are connected to ultrasonic sensors ESP8266, Servo SG90, and HC-SR04. The connection involves linking the servo data pin to the D3 pin of the NodeMcu. Subsequently, the echo pin is connected to D6, and the trigger pin is linked to D5, as depicted in Figure 20. Message queuing telemetry transport (MQTT) is used as an IoT communication protocol through the anto.io platform. In order to facilitate better comprehension and application, Figure 21 presents the hardware and software setup of an Ultrasonic Distance Sensor (UDS). Figure 21(a) shows the logic of the ultrasonic scanner moving from right to left and vice versa repeatedly with a maximum servo movement angle of 90°. Figure 21(b) shows the installation configuration between the servo motor and ultrasonic sensor, while Figure 22 shows the results of interfacing data from UDS.

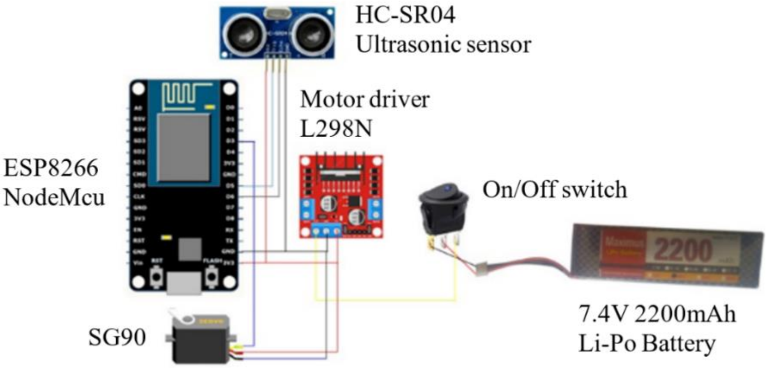
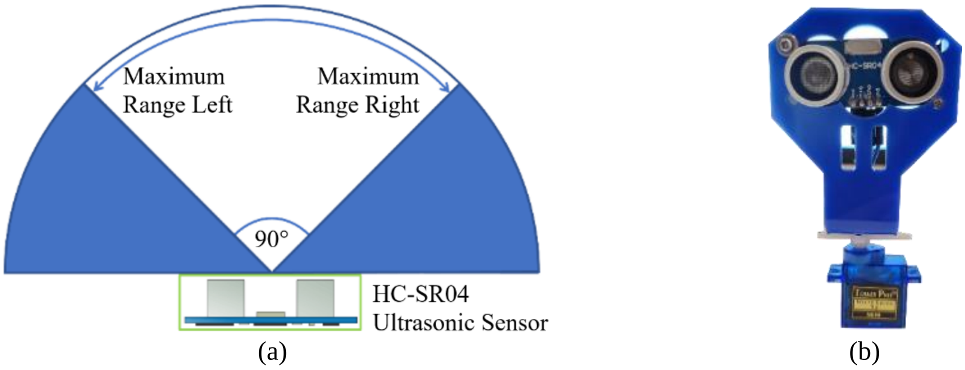


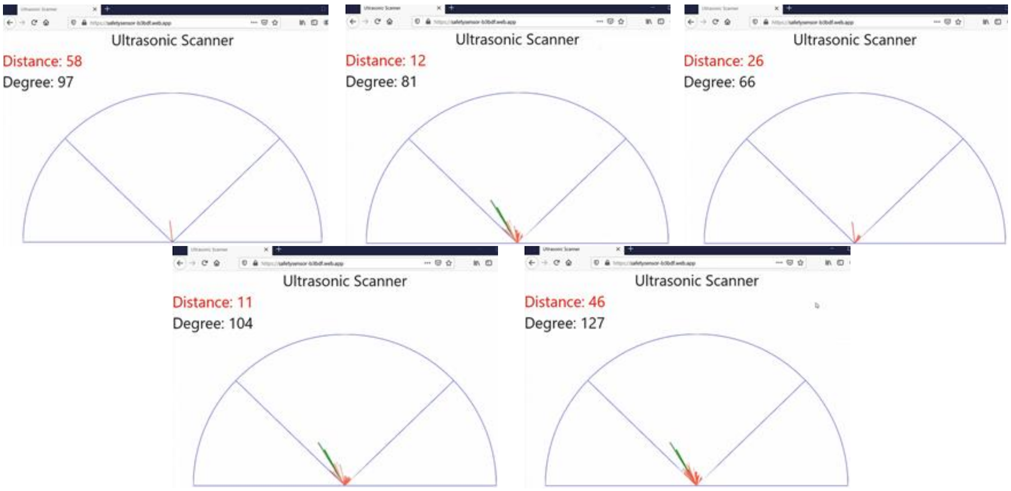
Figure 20. Wiring diagram of UDS



(a)

(b)

Figure 21. UDS configuration (a) software logic configuration and (b) hardware configuration



Ultrasonic Scanner
Distance: 58
Degree: 97

Ultrasonic Scanner
Distance: 12
Degree: 81

Ultrasonic Scanner
Distance: 26
Degree: 66

Ultrasonic Scanner
Distance: 11
Degree: 104

Ultrasonic Scanner
Distance: 46
Degree: 127

Figure 22. UDS system interface

### 3.6. Field testing

We carried out one field test with one track consisting of four destination points that were adjusted to each material delivery post in the factory. The waypoint path is planned to move counterclockwise. The waypoint coordinates are obtained based on the longitude and latitude readings on Google Maps. Figure 23 shows the results of AGV experiments in the field, showing the system's performance in following the path. WP1 is the starting point for the AGV, which then goes to WP2, WP3, and ends at WP4. The red line shows the planned waypoint, while the green dot and yellow line represent the actual response obtained from the AGV. It can be seen that the actual response and waypoint at several points have pretty significant differences. This is due to the resolution of GPS (Ublox Neo6M), where the error rate ranges from 4 meters to 7,756 meters (calculated based on longitude and latitude using the great circle distance formula). The GPS reading error is also affected by the test location being squeezed/obstructed by two buildings so that the signal reading is less than optimal. This discrepancy is also brought on by the compass sensor's (HMC5883L) readings varying, which is brought on by the AGV's vibration and irregular road conditions.

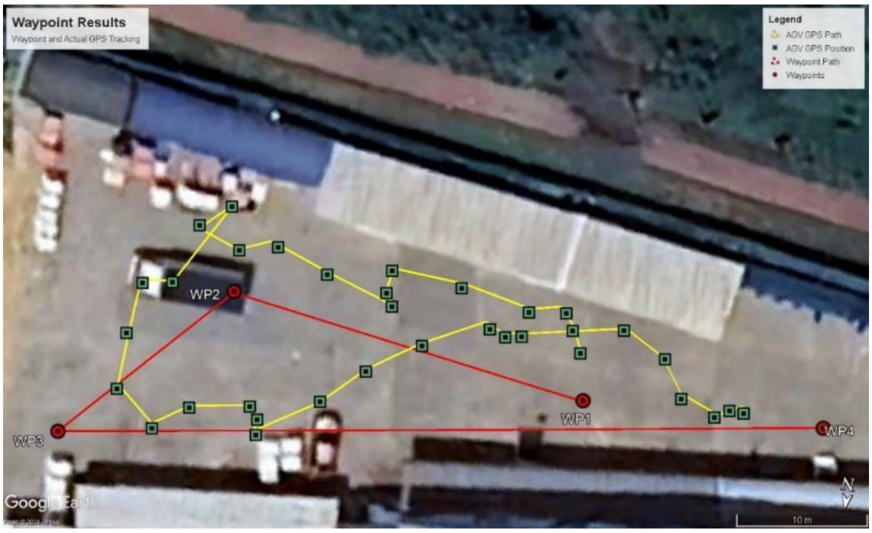


Figure 23. AGV testing on the factory

## 4. CONCLUSION

Pure pursuit control algorithm with obstacle avoidance for AGV simulation has been implemented. The simplified kinematic model of the AGV robot shows the effect of forwarding visibility values for PPA. The results show that a small look-ahead value has unwanted oscillations in the robot path. A range sensor was added to give the AGV the ability to avoid obstacles. Simulations are carried out using MATLAB/ Simulink with realistic environmental parameters. The combination of path following, and obstacle avoidance shows that the AGV robot can maneuver well. The latest research results show that the look-ahead value changes the robot's range to the reference point. In addition, no oscillations were observed in the motion of the robot. It can be said that the highest forward visibility has a positive effect on the robot's travel time. In this case, the robot changes its direction without approaching the reference point. In conclusion, the AGV field test exposed significant differences between planned and actual waypoints, with the GPS (Ublox Neo6M) showing an error rate from 4 to 7,756 meters. Signal obstruction and sensor fluctuations contributed to these discrepancies. To enhance precision, future efforts should focus on improving GPS accuracy, mitigating signal obstructions, and minimizing sensor fluctuations using sensors with better accuracy, such as real time kinematics global navigation satellite systems (RTK GNSS).

## ACKNOWLEDGEMENTS

We are deeply grateful to Rajamangala University of Technology Thanyaburi for their thoughtful and generous contribution. Their significant contribution played a crucial role in facilitating the completion of this scientific research paper. We sincerely thank them for their invaluable assistance.

## REFERENCES

[1] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, “Smart Factory of Industry 4.0: key technologies, application case, and challenges,” *IEEE Access*, vol. 6, pp. 6505–6519, 2018, doi: 10.1109/ACCESS.2017.2783682.

[2] S. Zhou, G. Cheng, Q. Meng, H. Lin, Z. Du, and F. Wang, “Development of multi-sensor information fusion and AGV navigation system,” in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, Jun. 2020, pp. 2043–2046, doi: 10.1109/ITNEC48623.2020.9084687.

[3] M. Mammarella, L. Comba, A. Biglia, F. Dabbene, and P. Gay, “Cooperative agricultural operations of aerial and ground unmanned vehicles,” in *2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, IEEE, Nov. 2020, pp. 224–229, doi: 10.1109/MetroAgriFor50201.2020.9277573.

[4] R. Bohrer, Y. K. Tan, S. Mitsch, A. Sogokon, and A. Platzer, “A formal safety net for waypoint-following in ground robots,” *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2910–2917, Jul. 2019, doi: 10.1109/LRA.2019.2923099.

[5] M. De Simone, Z. Rivera, and D. Guida, “Obstacle avoidance system for unmanned ground vehicles by using ultrasonic sensors,” *Machines*, vol. 6, no. 2, p. 18, Apr. 2018, doi: 10.3390/machines6020018.

[6] M. A. Kareem Jaradat, M. Al-Rousan, and L. Quadan, “Reinforcement based mobile robot navigation in dynamic environment,” *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 135–149, Feb. 2011, doi: 10.1016/j.rcim.2010.06.019.

[7] N. H. Singh and K. Thongam, “Mobile robot navigation using fuzzy logic in static environments,” *Procedia Computer Science*, vol. 125, pp. 11–17, 2018, doi: 10.1016/j.procs.2017.12.004.

[8] H. Khayyam, H. Ranjbarzadeh, and V. Marano, “Intelligent control of vehicle to grid power,” *Journal of Power Sources*, vol. 201, pp. 1–9, Mar. 2012, doi: 10.1016/j.jpowsour.2011.11.010.

[9] O. Castillo and P. Melin, “A review on interval type-2 fuzzy logic applications in intelligent control,” *Information Sciences*, vol. 279, pp. 615–631, Sep. 2014, doi: 10.1016/j.ins.2014.04.015.

[10] D. Aleksendrić, Ž. Jakovljević, and V. Ćirović, “Intelligent control of braking process,” *Expert Systems with Applications*, vol. 39, no. 14, pp. 11758–11765, Oct. 2012, doi: 10.1016/j.eswa.2012.04.076.

[11] P. Sudhakara, V. Ganapathy, B. Priyadharshini, and K. Sundaran, “Obstacle avoidance and navigation planning of a wheeled mobile robot using amended artificial potential field method,” *Procedia Computer Science*, vol. 133, pp. 998–1004, 2018, doi: 10.1016/j.procs.2018.07.076.

[12] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, “A review: on path planning strategies for navigation of mobile robot,” *Defence Technology*, vol. 15, no. 4, pp. 582–606, Aug. 2019, doi: 10.1016/j.dt.2019.04.011.

[13] F. Duchoň *et al.*, “Path planning with modified a star algorithm for a mobile robot,” *Procedia Engineering*, vol. 96, pp. 59–69, 2014, doi: 10.1016/j.proeng.2014.12.098.

[14] Y. Yang *et al.*, “An optimal goal point determination algorithm for automatic navigation of agricultural machinery: improving the tracking accuracy of the Pure Pursuit algorithm,” *Computers and Electronics in Agriculture*, vol. 194, p. 106760, Mar. 2022, doi: 10.1016/j.compag.2022.106760.

[15] Y. Jing, G. Liu, and C. Luo, “Path tracking control with slip compensation of a global navigation satellite system based tractor-scraper land levelling system,” *Biosystems Engineering*, vol. 212, pp. 360–377, Dec. 2021, doi: 10.1016/j.biosystemseng.2021.11.010.

[16] D. Maneetham, P. N. Crisnapati, and Y. Thwe, “Autonomous open-source electric wheelchair platform with internet-of-things and proportional-integral-derivative control,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, p. 6764, Dec. 2023, doi: 10.11591/ijece.v13i6.pp6764-6777.

[17] P. N. F. M. Shamsuddin, R. M. Ramli, and M. A. Mansor, “Navigation and motion control techniques for surface unmanned vehicle and autonomous ground vehicle: a review,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 1893–1904, Aug. 2021, doi: 10.11591/eei.v10i4.3086.

[18] P. N. Crisnapati, D. Maneetham, and E. Triandini, “Trolls: a novel low-cost controlling system platform for walk-behind tractor,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 1, p. 842, Feb. 2023, doi: 10.11591/ijece.v13i1.pp842-858.

[19] A. Jha and M. Kumar, “Two wheels differential type odometry for mobile robots,” in *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, IEEE, Oct. 2014, pp. 1–5, doi: 10.1109/ICRITO.2014.7014709.

[20] 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*, IEEE, Jun. 2011, pp. 27–32, doi: 10.1109/ICINFA.2011.5948958.

[21] 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.

[22] C. Mahulea, M. Kloetzer, and R. González, *Path Planning of Cooperative Mobile Robots Using Discrete Event Models*. Wiley, 2019. doi: 10.1002/9781119486305.

[23] A. Das, M. Ghosal, and D. Das, "Development and performance studies of electronic wheel slip control device for two wheel drive agricultural tractor," *International Journal of Chemical Studies*, vol. 8, no. 2, pp. 1575–1579, Mar. 2020, doi: 10.22271/chemi.2020.v8.i2x.8986.

[24] G. BOZTAŞ and Ö. AYDOĞMUŞ, "Implementation of Pure pursuit algorithm for nonholonomic mobile robot using robot operating system," *Balkan Journal of Electrical and Computer Engineering*, vol. 9, no. 4, pp. 337–341, Oct. 2021, doi: 10.17694/bajece.983350.

[25] G. Chen, N. Luo, D. Liu, Z. Zhao, and C. Liang, "Path planning for manipulators based on an improved probabilistic roadmap method," *Robotics and Computer-Integrated Manufacturing*, vol. 72, p. 102196, Dec. 2021, doi: 10.1016/j.rcim.2021.102196.

[26] J. C. Mohanta and A. Keshari, "A knowledge based fuzzy-probabilistic roadmap method for mobile robot navigation," *Applied Soft Computing*, vol. 79, pp. 391–409, Jun. 2019, doi: 10.1016/j.asoc.2019.03.055.

[27] T. Sutikno, H. Satrian Purnama, A. Pamungkas, A. Fadlil, I. Mohd Alsofyani, and M. H. Jopri, "Internet of things-based photovoltaics parameter monitoring system using NodeMCU ESP8266," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, p. 5578, Dec. 2021, doi: 10.11591/ijece.v11i6.pp5578-5587.

## BIOGRAPHIES OF AUTHORS

**Singhanart Ketsayom** completed the Bachelor of Industrial Technology Program in Production Technology from the Faculty of Technical Education at Rajamangala University of Technology Thanyaburi (RMUTT), Thailand, in 2018. Subsequently, in 2020, he enrolled in the master's program in Mechatronics Engineering within the same faculty at RMUTT. Currently serving as the Factory Manager at Kor. Rattanakit Co., Ltd, Singhanart brings a wealth of expertise in flexible robotics, along with unique skills in design plastic injection machinery and boiler system design. He can be contacted at email: singhanart_k@mail.rmutt.ac.th.

**Dechrit Maneetham** currently serves as an Associate Professor in Mechatronics Engineering at Rajamangala University of Technology in Thanyaburi, Thailand. His academic journey includes earning B.Tech. and M.S. degrees in Electrical and Computer Engineering from King Mongkut's University of Technology in North Bangkok. In 2010, he achieved a D.Eng. in Mechatronics from the Asian Institute of Technology, followed by a Ph.D. in Electrical and Computer Engineering from Mahasarakham University in 2018. With a wealth of experience exceeding 15 years in teaching engineering, he has significantly contributed to the field. His contributions extend to the publication of over 30 technical papers. Beyond academia, he has shared his knowledge through robotics short courses at various conferences and authored seven books. These publications cover a range of topics, including pneumatics system, hydraulics system, MCS-51 microcontroller, PIC microcontroller, Arduino microcontroller, PLC beckhoff, and robot. These works showcase his expertise in the realms of robotics, automation, mechatronics, and biomedical applications. He can be contacted at email: dechrit_m@rmutt.ac.th.

**Padma Nyoman Crisnapati** earned his bachelor's degree in 2009 from the Department of Informatics Engineering at Sepuluh Nopember Institute of Technology. Subsequently, he pursued a master's degree in Learning Technology in 2011 and another master's degree in computer science in 2018, both from Ganesha Education University. He successfully earned a D.Eng. in the Mechatronics Engineering Department at Rajamangala University of Technology Thanyaburi (RMUTT). His diverse research interests encompass various domains, including mechatronics, robotics, automation, augmented and virtual reality, 2D and 3D animation, and internet of things. He can be contacted at email: crisnapati@rmutt.ac.th.