

Research Article

Multicamera-Based Indoor Localization and Path Optimization for Mobile Robots Using ArUco Markers

Abdulhamit Sevgi^{1a}, Hasan Erdinç Koçer^{2b}

¹ Mechatronic Program, OSTİM Technical University Ankara, Türkiye

² Electrical & Electronics Engineering, Selçuk University, Konya, Türkiye

abdulhamit.sevgi@ostimteknik.edu.tr

DOI : 10.31202/ecjse.1614568

Received: 06.01.2025 Accepted: 07.04.2025

How to cite this article:

Abdulhamit Sevgi, Hasan Erdinç Koçer, "Multicamera-Based Indoor Localization and Path Optimization for Mobile Robots Using ArUco Markers", El-Cezeri Journal of Science and Engineering, Vol: 12, Iss: 3, (2025), pp.(339-355).

ORCID: "0000-0003-3567-848X"; ^b 0000-0002-0799-2140.

Abstract : In indoor environments where GPS signals are not available, determining the positions of mobile robots is challenging. Common methods include SLAM, marker-based localization, inertial measurement units (IMU), and hybrid positioning systems (HPS). SLAM enables simultaneous localization and mapping, while marker-based localization uses specific markers for position detection. IMUs track motion through velocity, acceleration, and angular velocity, and HPS combines sensors for improved accuracy. This study develops a route planning and motion optimization method using ArUco markers from the University of Córdoba. Detected via image processing, these markers guide robots by calculating the shortest and safest paths to the targets. Multiple cameras enhance motion range and vision, while an automatic pan adjustment addresses overlaps and alignment issues, ensuring seamless image integration. The proposed method demonstrates the potential of multicamera systems for reliable indoor navigation, offering promising applications in the industrial and service domains.

Keywords : Mobile Robots, Motion Optimization, Multi-Camera Systems, ArUco Markers, Indoor Localization.

1 Introduction

The localization and autonomous navigation of mobile robots have become pivotal areas of research, fostering the development of various systems and algorithms for position determination. Notable examples include Global Positioning System (GPS), Local Positioning Systems (LPS), and Hybrid Positioning Systems (HPS). GPS, which relies on satellite data to provide positional information, is widely used in outdoor settings as military operations, meteorology, vehicle monitoring, cartography, and agriculture. This system delivers precise data on location, velocity, and timing. However, GPS is not suitable for indoor use due to constraints such as signal strength, lack of line-of-sight, and diminished accuracy. In contrast, LPS can determine the location of objects in a Cartesian coordinate frame, which is particularly suitable for indoor environments [1], [2], [3]. It is an important component of electronic performance and monitoring systems that facilitate accurate and efficient data collection to monitor external loads. Techniques commonly associated with LPS include Wi-Fi, Bluetooth Low Energy (BLE), Radio Frequency Identification (RFID), and Ultra Wideband (UWB). Although LPS offers advantages in terms of precision, power efficiency and indoor coverage, it is more complex and costly than GPS due to the need for dedicated infrastructure and hardware deployment. Hybrid Positioning Systems (HPS) integrate GPS and LPS technologies to provide seamless tracking in both indoor and outdoor environments. Using multiple data sources, HPS overcomes the limitations of GPS in confined or urban areas, improves positional accuracy, and expands the applicability of the system in various environments [4]. Localization systems are generally classified as outdoor and indoor positioning. For outdoor environments, GPS is preferred as a low-cost solution with high accuracy. In contrast, for indoor environments where GPS is ineffective, Indoor Positioning Systems (IPS) are essential. IPS uses a variety of technologies, including magnetic sensors, inertial sensors, acoustic sensors (audible, ultrasonic, and acoustic sounds), optical sensors, radio frequency-based sensors, and image processing with marker-based systems [3].

Image processing-based systems are based on the processing and analyzing of video data, employing fixed and mobile camera systems for localization [5]. Fixed camera systems use stationary cameras to track moving objects, determining their positions based on prominent features in the camera's field of view. Mobile camera systems, equipped with localization

hardware, involve two stages: an offline phase where images of the environment are captured and processed to extract unique features, and an online phase where captured images are matched with stored features to determine the camera position.

Modern mobile robots are often equipped with onboard camera systems [6], providing an effective tool for determining their position within the environment. Visual localization is a subset of trigonometric methods, where vision systems consisting of one or two cameras detect markers using computer vision techniques [7], [8], [9]. Robots can derive information about the physical characteristics of the environment using these tools combined with various algorithms. For single-camera systems, the robot's position is calculated on the basis of marker positions in the environment captured during movement. Determining the robot's pre-movement coordinates requires knowledge of at least two markers, although static robot positions can be identified relative to a single marker, particularly when specialized 2D markers are used. Knowing the calibration parameters of the camera lens allows one to calculate the displacement and rotation of the marker in the image plane.

Several types of 2D marker systems have been developed for augmented reality applications, including ARStudio, ARToolkit, and ARTag [10]. These systems utilize plane patterns recognized by algorithms and ensure that marker patterns cannot be created by rotating another used pattern in the environment. These requirements are met by the ArUco marker system, which includes a library of functions for detection and localization. The monochromatic design of ArUco markers allows for efficient and rapid detection within a frame, with minimal computational effort. Visual localization using reference markers has been employed in UAV navigation and landing algorithms [11], [12] and mobile robot localization [13]. In addition, studies have explored landmark-based navigation, focusing on feature extraction and recognition of static environmental characteristics [14].

In this study, a localization and motion optimization algorithm is developed based on data obtained from ArUco markers. The algorithm calculates efficient routes to predefined targets, enabling the effective and reliable movement of mobile robots in indoor environments. The results of the proposed approach have been observed and analyzed.

2 Experimental Methods

2.1 ArUco Tags

ArUco markers are widely used in the field of computer vision for target detection, pose estimation, and position tracking. In this study, 5×5 markers are used, with markers numbered 00–05 designated for vehicles and markers numbered 11–17 assigned as targets. Identifying the pixel coordinates of all markers allows one to accurately determine the real-world positions of the vehicles within the local coordinate system. Figure 1 provides a visual representation of the ArUco codes used in this study.

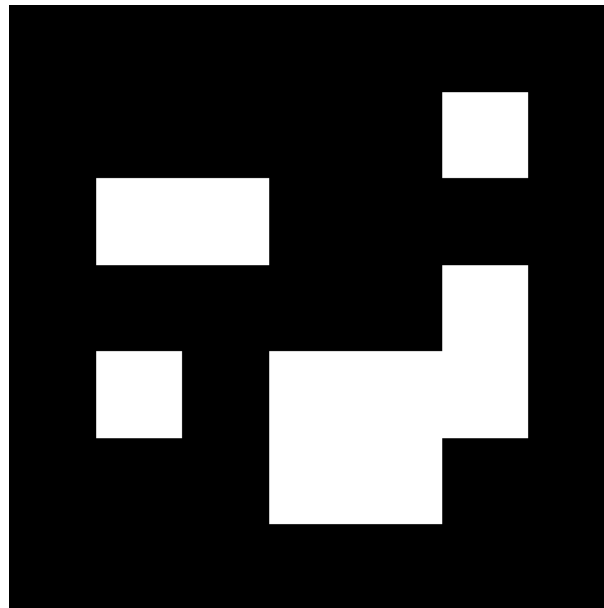


Figure 1: ArUco Markers.

The perspective transformation is used to determine the positions of vehicles [15]. By performing a perspective transformation using the pixel coordinates of the markers and their corresponding world coordinates, the two-dimensional information of the markers can be directly obtained. Since the pixel coordinates are proportional to the world coordinates, simple operations on the pixel coordinates enable the calculation of both the world coordinates and the orientation of the vehicles. In

this study, ArUco markers are utilized to identify the positions of the four edges of robots and targets.

2.2 System Components

The robot developed for this study has three wheels: two active drive wheels located at the rear and one swiveling caster wheel in the front. It is designed with dimensions of $140 \times 140 \times 120$, mm (Width $\times$ Length $\times$ Height) to accommodate all components and is manufactured using a 3D printer. The structural parameters are detailed in the mathematical models. Figure 2 illustrates one of the mobile robots produced.

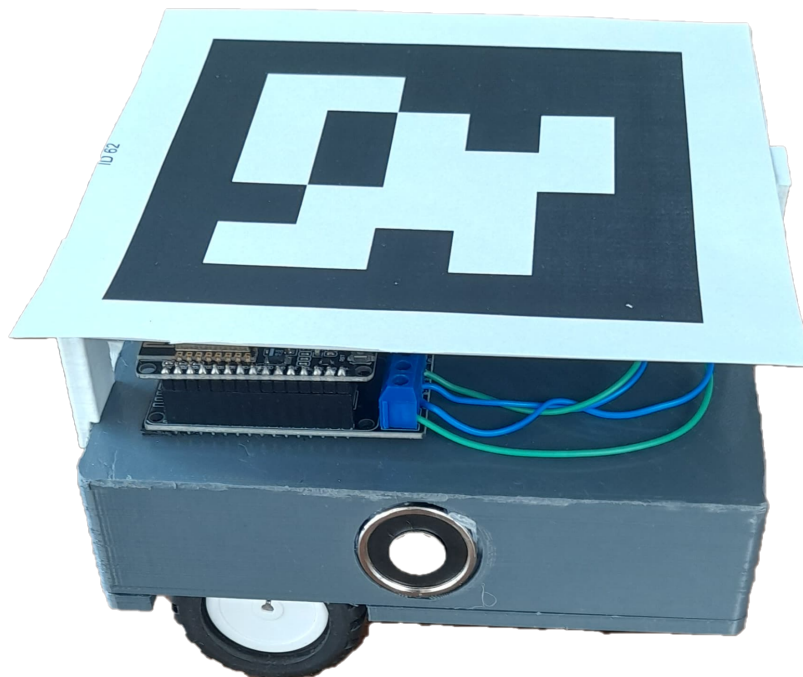


Figure 2: Produced Mobile Robot.

Communication within the system is facilitated by the ESP8266 controller board. These devices utilize the Wi-Fi communication protocol, which enables operation in low-power wireless networks. All robots and the main controller are connected to a single Wi-Fi network and are assigned static IP addresses. The other electronic components of the system are illustrated in Figure 3.

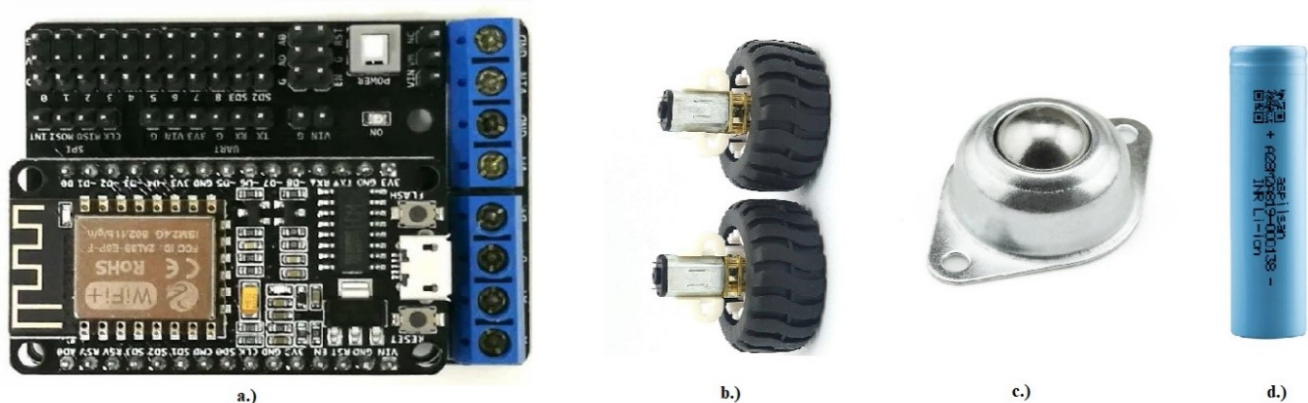


Figure 3: System Components: (a) ESP8266 and L293D motor driver board, (b) N20 motor and wheel, (c) Swivel caster wheel, (d) LiPo battery.

2.3 Kinematic Model of the Robot

Robots are controlled and classified according to their movement mechanisms. The motion of a robot is influenced by several factors, including maximum speed, total mass, wheel structure, differential mechanisms, and other variables. Although the precise interaction between load-bearing skeletal elements can be complex, joint behavior is typically estimated using linear and rotational kinematic joints. In this study, we plan the design and implementation of a two-wheeled mobile robot with differential drive, following the kinematic and dynamic simulation models available in the literature. Velocity decomposition is employed to address the robot's kinematic model. The velocity distributions of the right wheel and the robot body are illustrated in Figure 4.

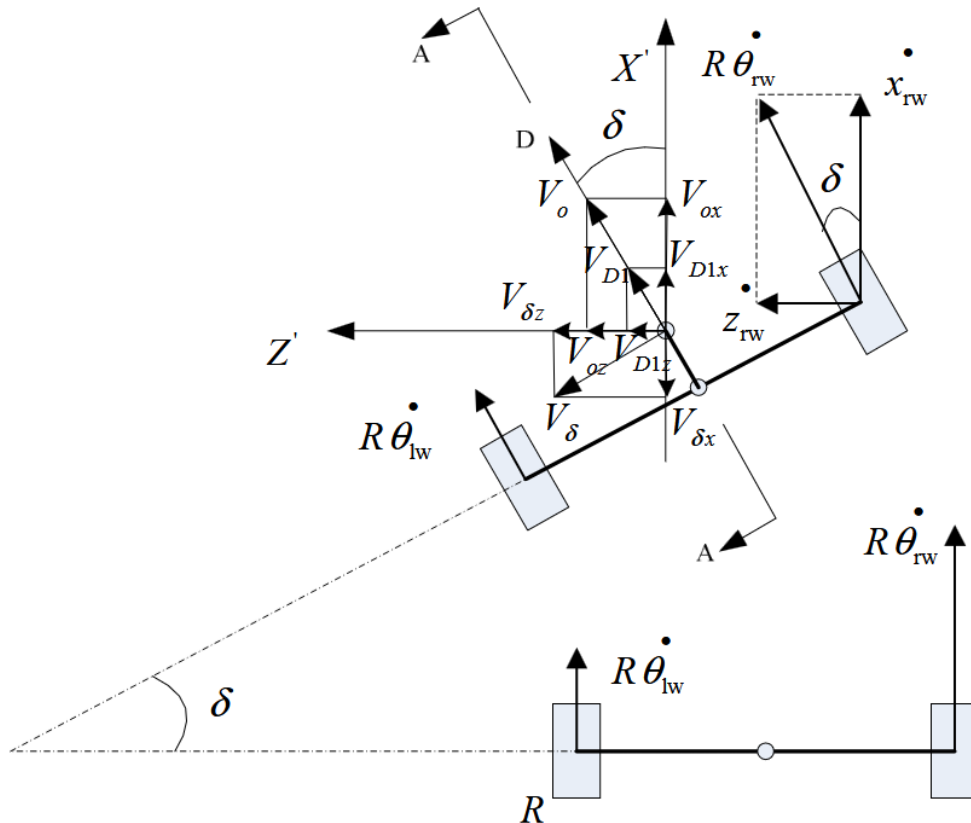


Figure 4: Velocity distributions of the robot [16].

In this robot, the vertical axis that passes through the midpoint of the two wheels represents the positive Y -axis, the direction of the robot's movement indicates the positive X -axis, and the axis of the two wheels corresponds to the positive Z -axis. The cross-sectional view (AA) of the robot is illustrated in Figure 5.

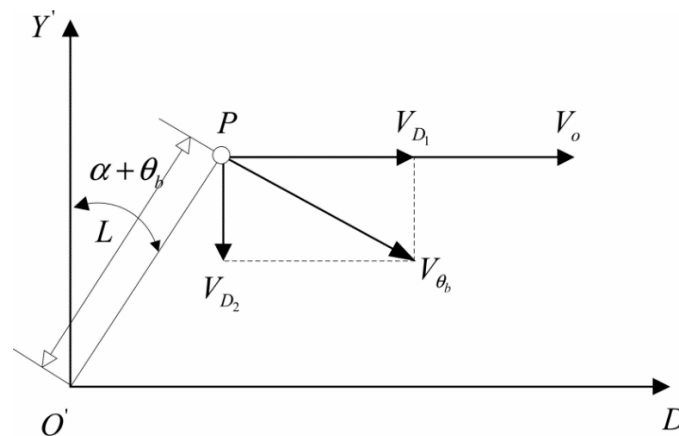


Figure 5: AA Cross-Sectional View of the Robot [16].

From Figures 4 and 5, the following parameters are defined:

- $\dot{R}_{rw}$: Linear velocity of the right wheel (m/s).
- $\dot{x}_{rw}$: Velocity of the right wheel along the X -axis (m/s).
- $\dot{z}_{rw}$: Velocity of the right wheel along the Z -axis (m/s).
- δ : Deviation angle between the robot's current direction and its initial direction (rad).
- $\dot{\theta}_{rw}$: Angular velocity of the right wheel (rad/s).
- R : Radius of the wheel (m).
- $\dot{\theta}_{lw}$: Angular velocity of the left wheel (rad/s).
- V_{θ_b} : Rotational velocity of the center of mass around the OX axis (rad/s).
- L : Distance between the center of mass and the axle of the two wheels (m).
- V_{D2} : Velocity of V_{θ_b} along the OY axis (m/s).
- V_{D1} : Velocity of V_{θ_b} along the OD axis (m/s).
- V_o : Velocity of the robot's chassis center (m/s).
- V_{δ} : Rotational velocity of the center of mass around the OY axis (m/s).

The velocity analysis of the right wheel, as illustrated in Figure 4, is presented in Equation 2:

$$\begin{aligned}\dot{x}_{rw} &= R\dot{\theta}_{rw} \cos \delta, \\ \dot{y}_{rw} &= 0, \\ \dot{z}_{rw} &= R\dot{\theta}_{rw} \sin \delta.\end{aligned}\tag{1}$$

Using the same method, the velocity of the left wheel is calculated as follows:

$$\begin{aligned}\dot{x}_{lw} &= R\dot{\theta}_{lw} \cos \delta, \\ \dot{y}_{lw} &= 0, \\ \dot{z}_{lw} &= R\dot{\theta}_{lw} \sin \delta.\end{aligned}\tag{2}$$

Equations 1 and 2 define the velocity components of the right and left wheels. The velocity equation for the right wheel decomposes the linear velocity at the wheel's contact point with the ground into two components. The velocity component along the X -axis is calculated by multiplying the angular velocity by the radius of the wheel and projecting it onto the horizontal plane based on the robot orientation angle. Similarly, the velocity component along the Z -axis is derived by multiplying the angular velocity by the radius of the wheel and determining its projection onto the vertical plane according to the orientation of the robot. The velocity along the Y -axis is assumed to be zero, as the robot operates strictly within the horizontal plane. The equations for the left wheel follow the same principles as those for the right wheel, but they account for the independent angular velocities of each wheel.

V_{D1} , V_o , and V_{δ} are decomposed along the X -axis as V_{D1x} , V_{ox} , and $V_{\delta x}$, along the Z -axis as V_{D1z} , V_{oz} , and $V_{\delta z}$, and along the Y -axis as V_{D1y} , V_{oy} , and $V_{\delta y}$. The velocity equation along the X -axis is:

$$\begin{aligned}V_{D1x} &= V_{D1} \cos \delta, \\ V_{D1} &= V_{\theta_b} \cos(\alpha + \theta_b), \\ V_{\theta_b} &= L(\alpha + \theta_b).\end{aligned}\tag{3}$$

$$\begin{aligned}V_{ox} &= V_o \cos \delta, \\ V_o &= \frac{R\dot{\theta}_{lw}^* + R\dot{\theta}_{rw}^*}{2}.\end{aligned}\tag{4}$$

$$\begin{aligned}V_{\delta x} &= -V_{\delta} \sin \delta, \\ V_{\delta} &= \dot{\delta} L \sin(\alpha + \theta_b), \\ \dot{\delta} &= \frac{\dot{\theta}_{lw} - R\dot{\theta}_{rw}}{D}.\end{aligned}\tag{5}$$

To determine the total velocity of the robot's body along the X -axis, various velocity components are combined. The first component represents the projection of linear motion onto the X -axis, which varies according to the orientation angle. The second component corresponds to the velocity of the center of the chassis, projected on the basis of the orientation angle. Lastly, the third component arises from the rotational motion of the robot's body center and contributes to the velocity along the X -axis. The total velocity along the X -axis is expressed as the sum of these three components, modeling the combined effects of

linear motion and rotational dynamics. Following these equations, the decomposed velocities along the X , Y , and Z axes are integrated to establish the complete kinematic model of the robot, ensuring consistency between the wheel velocities and the overall motion of the robot. This unified approach simplifies the analysis and facilitates accurate motion control.

$$\begin{aligned} V_x &= V_{D1x} + V_{ox} + V_{\delta x} \\ &= L(\alpha + \theta_b) \cos(\alpha + \theta_b) \cos \delta + \frac{R(\dot{\theta}_{lw} + \dot{\theta}_{rw})}{2} \cos \delta \\ &\quad - \frac{R(\dot{\theta}_{lw} - \dot{\theta}_{rw})}{D} L \sin(\alpha + \theta_b) \sin \delta. \end{aligned} \quad (6)$$

These relationships are established using the appropriate formula.

The velocity along the Y -axis is expressed as:

$$V_y = V_{D2} = -V_{\theta b} \sin(\alpha + \theta_b) = -L(\alpha + \dot{\theta}_b) \sin(\alpha + \theta_b). \quad (7)$$

The velocity component of the robot's body along the Y -axis is derived by considering the effects of both inclination and body angle. The first term represents the linear velocity component along the Y -axis, which is influenced by the inclination angle and the tilt of the body. The second term represents the velocity component on the Y -axis of the center of the chassis as the robot moves. This equation helps to understand how the vertical movement of the center of the body is affected when the robot operates on an inclined surface.

The velocity along the Z -axis is expressed as:

$$\begin{aligned} V_{D1z} &= V_{D1} \sin \delta, \\ V_{D1} &= V_{\theta b} \cos(\alpha + \theta_b), \\ V_{\theta b} &= L(\alpha + \dot{\theta}_b). \end{aligned} \quad (8)$$

$$\begin{aligned} V_{oz} &= V_o \sin \delta, \\ V_o &= \frac{R\dot{\theta}_{lw} + R\dot{\theta}_{rw}}{2}. \end{aligned} \quad (9)$$

$$\begin{aligned} V_{\delta z} &= V_{\delta} \cos \delta, \\ V_{\delta} &= \dot{\delta} \cdot L \sin(\alpha + \theta_b), \\ \dot{\delta} &= \frac{R\dot{\theta}_{lw} - R\dot{\theta}_{rw}}{D}. \end{aligned} \quad (10)$$

The overall velocity along the Z -axis is:

$$\begin{aligned} V_Z &= V_{D1z} + V_{oz} + V_{\delta z} \\ &= L(\alpha + \dot{\theta}_b) \cos(\alpha + \theta_b) \sin \delta + \frac{R(\dot{\theta}_{lw} + \dot{\theta}_{rw})}{2} \sin \delta \\ &\quad + \frac{R(\dot{\theta}_{lw} - \dot{\theta}_{rw})}{D} L \sin(\alpha + \theta_b) \cos \delta. \end{aligned} \quad (11)$$

The total velocity of the robot along the Z -axis is determined by integrating the linear velocity, the rotational velocity around its center, and additional components influenced by the orientation angle. First, the projection of linear motion onto the Z -axis is calculated. Second, the velocity component resulting from the robot's rotational motion about its center is considered along this axis. Third, an additional velocity component is included to account for changes in the orientation angle. These equations illustrate how the robot moves along the Z -axis while changing direction, elucidating the shifts and orientation adjustments that occur during rotation. Based on the provided kinematic equations, the rotational angles of the robot's right and left wheels are presented in Equation 12:

$$\begin{aligned} \text{Left Angle} &= \phi = \theta_{lw} - \theta_b, \\ \text{Right Angle} &= \varphi = \theta_{rw} - \theta_b. \end{aligned} \quad (12)$$

These equations illustrate the relationship between the angular motions of the robot's left and right wheels and the tilt angle of its body. The difference in angular displacement between the two wheels determines the relative inclination of the body. As the angular movement of the body is influenced by the independent motion of the wheels, the body stability fluctuates based on

the interaction of these variables. After determining the angle values, the center point of the robot's chassis and the deviation angle can be calculated using Equations 13 and 14, respectively:

$$p = \frac{p_{lw} + p_{rw}}{2} = \frac{R\theta_{lw} + R\theta_{rw}}{2} = \frac{R}{2}(\phi + \varphi) + R(\alpha + \theta_b). \quad (13)$$

This equation calculates the travel distance of the robot. The total displacement of the robot is influenced by the combined angular motions of both the right and left wheels, as well as variations in the body's tilt angle. By multiplying the total angular movement of the wheels by the radius of the wheel, the total forward movement of the center of the chassis is determined. Furthermore, on inclined surfaces or under unbalanced conditions, an additional displacement component resulting from the body's angular tilt is included in the equation.

$$\delta = \frac{p_{lw} - p_{rw}}{D} = \frac{R\theta_{lw} - R\theta_{rw}}{D} = \frac{R}{D}(\phi - \varphi). \quad (14)$$

This equation determines the robot's change in direction. The variation in the robot's orientation angle is calculated on the basis of the difference in angular motion between the right and left wheels. One of the key parameters influencing the turning angle is the distance between the two wheels, known as the wheelbase width. If both wheels rotate at the same speed, the robot moves in a straight line; however, if one wheel rotates faster than the other, the robot changes direction along a curved path. This equation facilitates the analysis of the robot's turning behavior on the basis of its wheel speeds. These equations illustrate that, by using the robot's kinematic model, the desired motion can be accomplished by controlling the velocities of the two wheels [16].

After deriving the kinematic models of the robots, one of the most critical steps in motion planning is the development of a navigation system. In this study, the navigation system comprises two video cameras and ArUco markers, as illustrated in Figure 6.

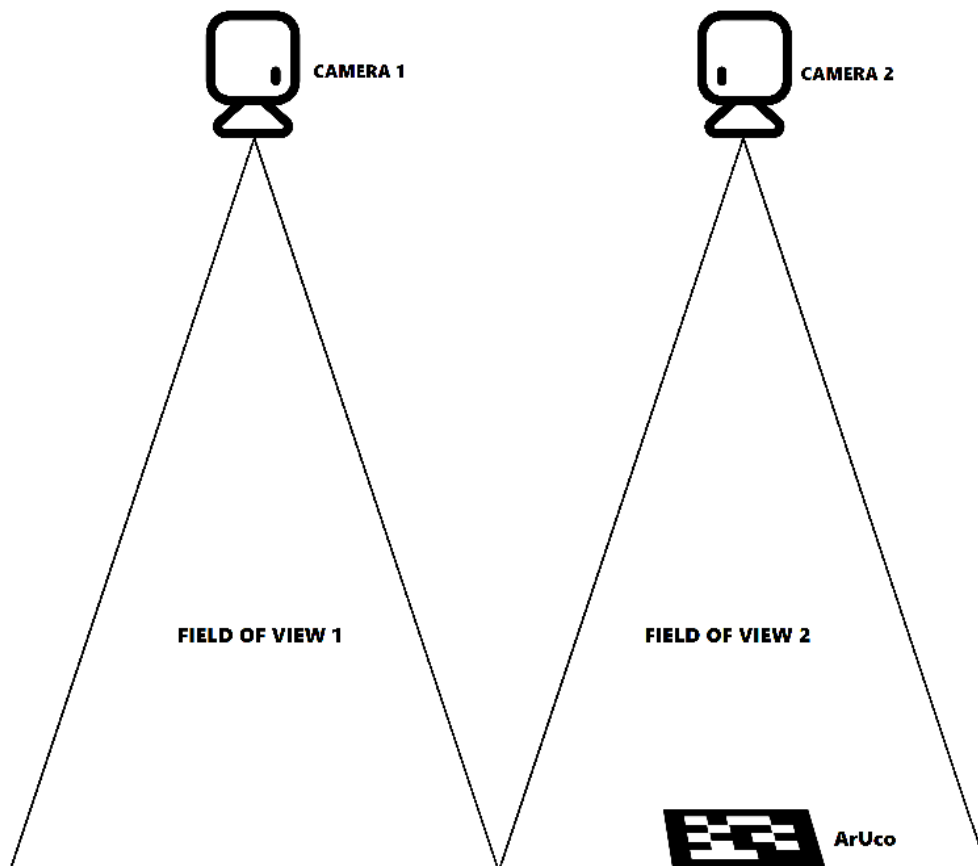


Figure 6: Cameras and System Field of View.

To track the robots, two Logitech C505e cameras are used to merge their fields of view. During the image merging process, the overlap between the fields of view of the cameras is identified, and any deviations in camera panning are corrected. In this

context, feature points are extracted using the Scale-Invariant Feature Transform (SIFT) algorithm. The percentage of overlap is calculated and this information is used to ensure the precise alignment of the cameras [17]. Given that the cameras are mounted at a height of 3 meters, the operational area they can capture measures $3 \times 6 \text{ m}^2$ (Figure 7). This area is organized to accommodate multiple robots, each equipped with a unique matrix reference marker.



Figure 7: System Workspace.

The size of the ArUco markers is 5×5 . Python programming language, in conjunction with OpenCV libraries, is used to process the ArUco markers. By placing an ArUco marker on top of the robot, aligned with its center of mass, the robot's position and orientation can be accurately determined. ArUco markers are square-shaped and feature a unique binary matrix along with a border for each marker, as shown in Figure 1. This internal matrix consists of ones and zeros, which define the unique identifier of the marker—black squares represent 0, while white squares represent 1, as illustrated in Figure 8. The marker border is black and is equivalent to one bit in width. Of the 100 possible markers that the library can generate, 20 are utilized in this study.

0	0	0	0	0	0	0
0	1	0	1	1	1	0
0	1	1	0	0	1	0
0	0	1	1	1	0	0
0	1	1	1	0	0	0
0	0	0	1	1	0	0
0	0	0	0	0	0	0

Figure 8: Numbered and Segmented Visualization of Black and White Squares.

Before detecting markers, calibration is performed to extract camera parameters. This process utilizes a chessboard pattern for camera calibration. A large chessboard with dimensions of 8×6 squares and an inner square size of 35 mm is used. To ensure accurate calibration, the chessboard is positioned at various locations within the camera frame, as illustrated in Figure 9. Upon completion of this process, the camera calibration matrix and distortion coefficients are calculated. The calibration matrix includes the focal lengths of the camera lens and the position of the projection center. These parameters facilitate the translation of the 2D coordinates of an object in the captured image into its corresponding 3D coordinates in the real world. Furthermore, the distortion coefficients correct for the geometric distortions introduced by the camera optics in the captured images.

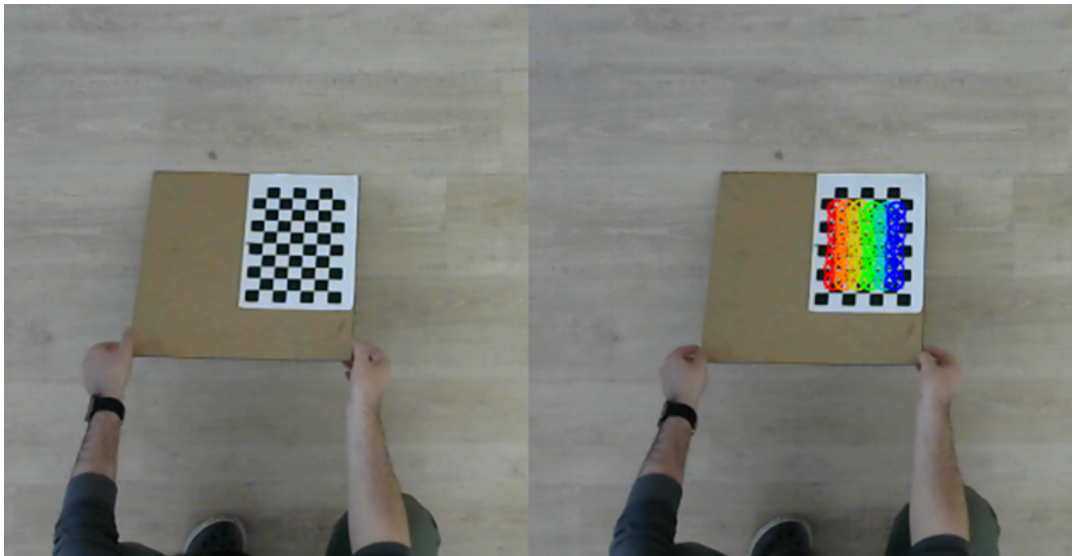


Figure 9: Calibration Process Using a Chessboard.

For a camera mounted at a height of 3 meters, a resolution of 1280×720 is adequate to track markers within the workspace. At this resolution, the numerical identifier of each marker is clearly visible, with the X-axis represented in red and the Y-axis in green, as illustrated in Figure 10. Identification of an ArUco tag involves several transformation steps applied to the original image. Initially, the image was converted to grayscale. Next, a binarization process is performed in which each pixel is converted to black or white based on its intensity. Finally, a contour analysis is performed to identify and isolate objects with a defined pattern. The identified marker is rendered in white, while the background remains black.

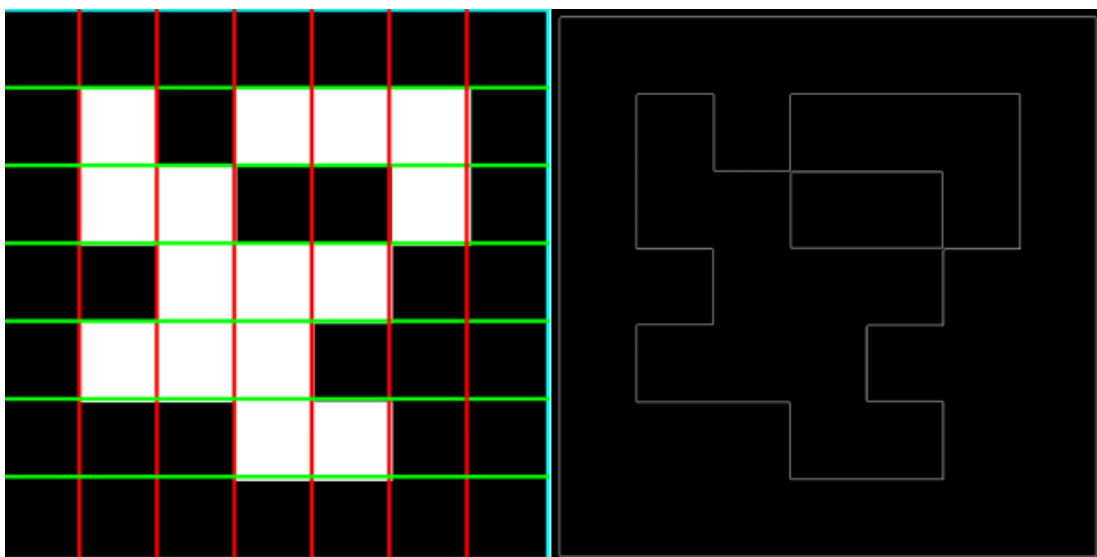


Figure 10: Segmented ArUco with Vertical-Horizontal Gridlines and Identified Borders.

Subsequently, polygon approximations are derived from the identified contours, and a predefined 5×5 grid is generated for the binary encoding analysis, as illustrated in Figure 11. In this grid, the black and white areas are assigned values of 0 and 1, respectively, enabling the determination of the marker's ID in matrix format.

1	0	1	1	1
1	1	0	0	1
0	1	1	1	0
1	1	1	0	0
0	0	1	1	0

Figure 11: Visualization with 5×5 Grid Applied.

Using this matrix, the system can determine the center position and orientation of the marker relative to the working plane, as illustrated in Figure 12.

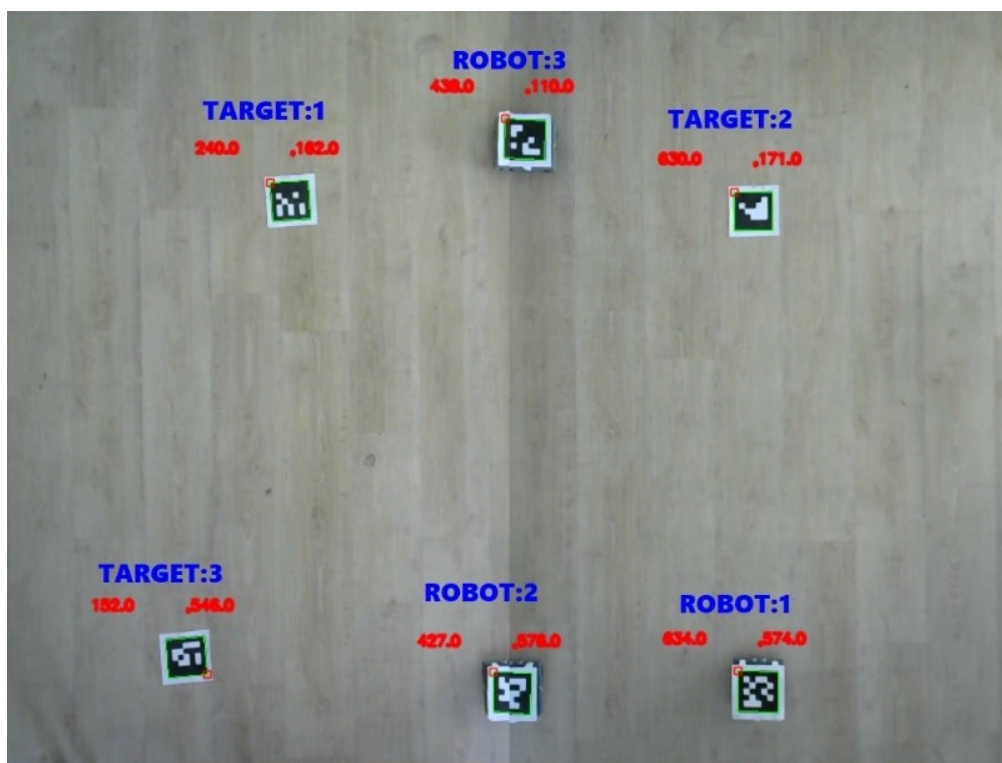


Figure 12: Positioning of Robots and Targets Using ArUco Codes by Cameras.

The pixel-based positional data of the center points of the identified robots and targets have been obtained. Using this information, the distance between the robots and their targets can be calculated employing the Euclidean formula:

$$\text{Distance} = \sqrt{(x_{\text{target}} - x_{\text{robot}})^2 + (y_{\text{target}} - y_{\text{robot}})^2}. \quad (15)$$

Here, x_{robot} and y_{robot} represent the coordinates of the current position of the robot, while x_{target} and y_{target} denote the coordinates of the target position. This calculation determines the linear distance from the robot's current position to the target position. The orientation angle of the robot relative to the target is calculated as follows:

The vector $\mathbf{r}_1$, representing the orientation towards the target, is defined as:

$$\mathbf{r}_1 = (x_{\text{target}} - x_{\text{current}}, y_{\text{target}} - y_{\text{current}}). \quad (16)$$

The vector $\mathbf{r}_2$, representing the robot's current orientation, is defined as:

$$\mathbf{r}_2 = (x_{\text{front}} - x_{\text{current}}, y_{\text{front}} - y_{\text{current}}). \quad (17)$$

The closure angle θ between the two vectors $\mathbf{r}_1$ (orientation towards the target) and $\mathbf{r}_2$ (current orientation of the robot) is calculated using the dot product formula:

$$\theta = \arccos \left(\frac{\mathbf{r}_1 \cdot \mathbf{r}_2}{\|\mathbf{r}_1\| \|\mathbf{r}_2\|} \right). \quad (18)$$

Here, x_{front} and y_{front} represent the position coordinates of the front part of the robot. The calculated angular error indicates the necessary angular correction to enhance the robot's orientation accuracy. If the angle is expressed in degrees:

$$\text{Angular Error} = \theta \times \frac{180}{\pi}. \quad (19)$$

These calculations have been used to determine the positional and orientation errors required for the robot to accurately and efficiently reach the target position. Once the robots' positioning within the system is finalized, communication between the main controller and the robots is established through Wi-Fi signals. The general operational block diagram of the system is illustrated in Figure 13.

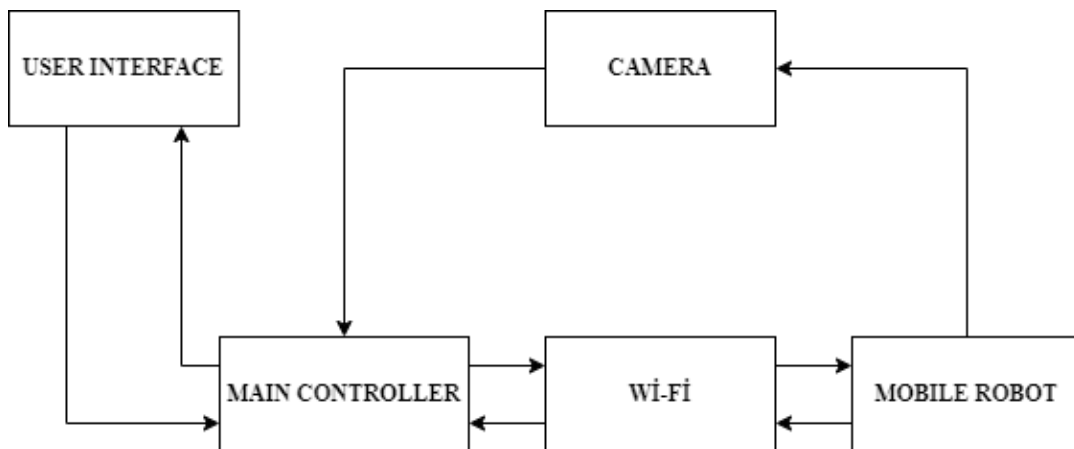


Figure 13: System Block Diagram.

As illustrated in the block diagram, the positions of the robots and targets are determined by the main controller using data from the cameras. The user interface allows the operator to select which robot should approach which target. Subsequently, movement commands are transmitted to mobile robots via the Wi-Fi communication protocol. An internal speed control mechanism has been designed and implemented for each mobile robot, utilizing two PID controllers, one for each wheel. This setup enables speed adjustments based on the distance and angular differences between the robots and their targets, ensuring that the robots accurately reach the desired positions. The flow chart of the system's operational algorithm is presented in Figure 14

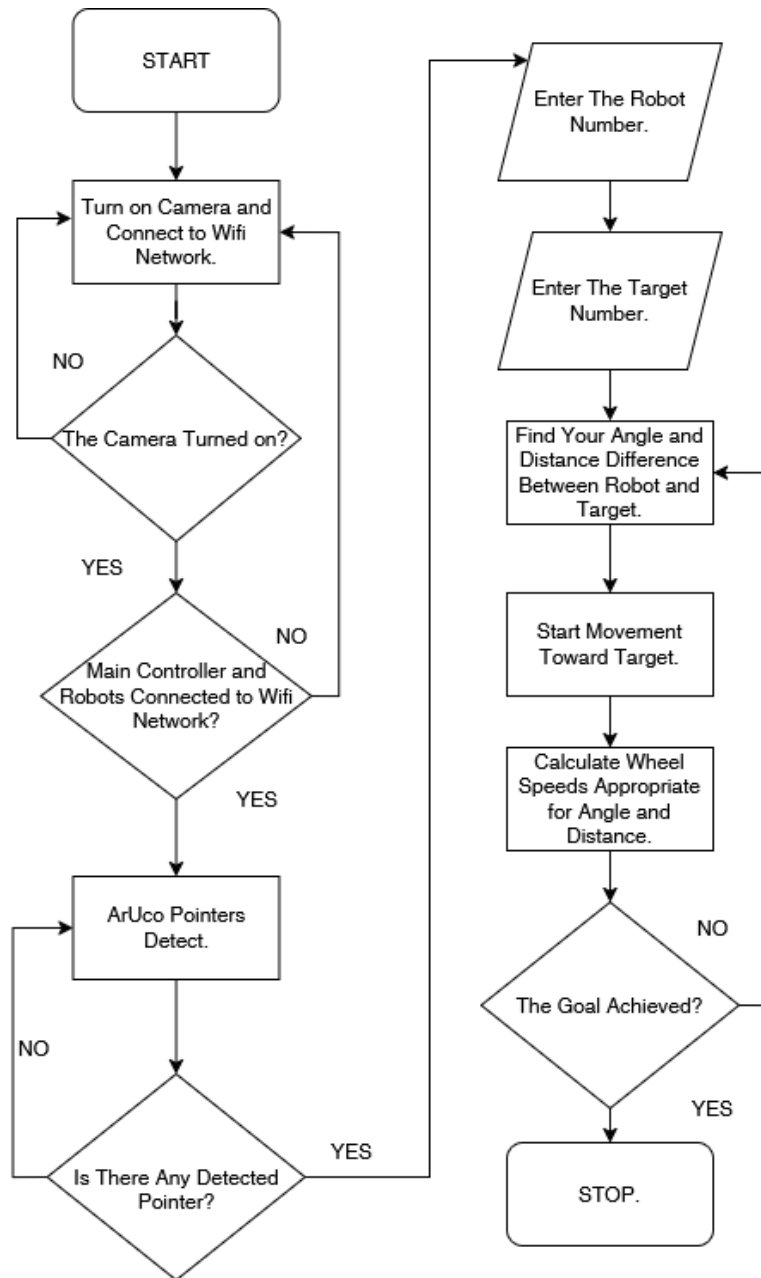


Figure 14: Algorithm Flowchart.

3 Results and Discussion

Experimental studies are conducted to evaluate the accuracy of robots in reaching predefined targets. In this context, the movement of the selected robot towards a designated target, as specified through the user interface, is examined in detail. The graphical user interface (GUI), developed using Python, significantly enhances the execution, monitoring, and control of the experiments. The system identifies ArUco markers in the environment, detects their positional information, and records the data on a centralized main controller. The data transmitted to the main controller integrate seamlessly with the robots' control software, ensuring continuous access to the position and orientation information of the markers.

During the experiment, the distances and angular differences between the robots and their targets are analyzed in detail. In Figure 12, a real-time snapshot of three different robots and their designated targets. Through these images, robots' positions, orientations, and distances from their targets are continuously monitored and analyzed. Based on these data, speed and directional adjustments are made according to the robots' kinematic models, ensuring that each robot reaches its target efficiently. Once the robots successfully reach their targets, the system marks the tasks as completed, concluding the process. Such systems demonstrate significant potential, particularly in fields such as industrial robotics and autonomous vehicles. In this study, two

experimental scenarios are designed. The first scenario involves one robot and three targets within the field of view of the camera. This experiment is carried out 30 times for each robot, resulting in a total of 90 trials. A snapshot of the robot reaching the first target is presented in Figure 15.

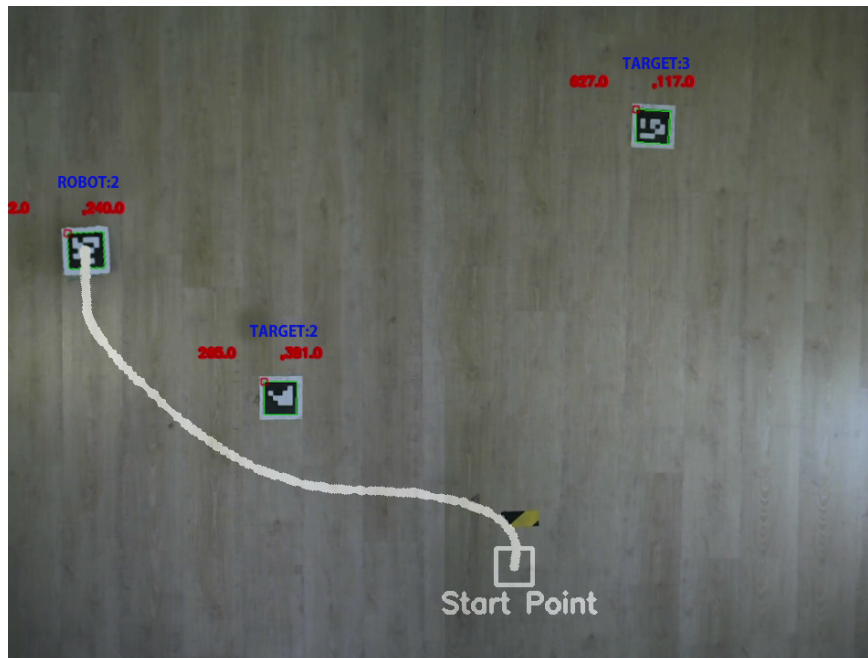


Figure 15: Experimental Scenario Where the Robot Reaches the First Target.

In the second experimental scenario, three robots are programmed to move toward three distinct targets. During this phase, a total of 30 trials are conducted. The robots are initially directed toward linear targets and then toward lateral targets in a mixed sequence, and the results are recorded. In the experimental setup that involves multiple robots and targets in the field, the robots are dispatched sequentially to their designated targets. The experimental results are illustrated in Figure 16.

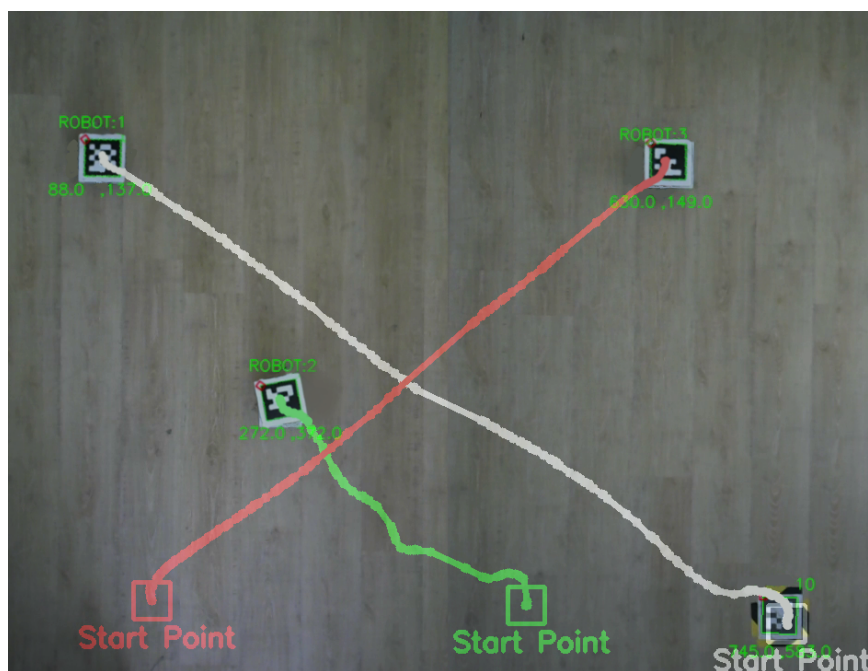


Figure 16: Experimental Scenario Where Robots Move Sequentially Toward and Reach Their Targets.

The developed positioning system effectively ensures that robots reach their targets in all experimental trials. Compared to other positioning systems documented in the literature, the proposed system is both cost-effective and capable of delivering more precise results. The findings of the experimental study are presented in Table 1.

Table 1: Error Rates of Experimental Studies (mm).

Experiment	Single Robot Experiments			Collective Robots		
	Robot 1	Robot 2	Robot 3	Robot 1	Robot 2	Robot 3
1	5.00	2.00	11.00	14.00	13.00	19.00
2	8.00	5.00	13.00	9.00	12.00	10.00
3	10.00	7.00	5.00	8.00	13.00	15.00
4	4.00	5.00	2.00	5.00	13.00	18.00
5	5.00	4.00	6.00	16.00	7.00	19.00
6	6.00	5.00	7.00	18.00	14.00	18.00
7	3.00	12.00	8.00	8.00	12.00	13.00
8	16.00	3.00	2.00	6.00	17.00	18.00
9	11.00	8.00	5.00	1.00	10.00	16.00
10	4.00	5.00	7.00	16.00	8.00	15.00
11	13.00	11.00	11.00	6.00	17.00	18.00
12	13.00	14.00	2.00	16.00	14.00	17.00
13	5.00	5.00	4.00	8.00	21.00	17.00
14	2.00	12.00	11.00	11.00	14.00	11.00
15	6.00	11.00	11.00	14.00	13.00	19.00
16	13.00	6.00	12.00	5.00	13.00	18.00
17	8.00	6.00	8.00	16.00	8.00	15.00
18	2.00	4.00	6.00	16.00	14.00	17.00
19	11.00	2.00	5.00	8.00	12.00	13.00
20	7.00	7.00	5.00	18.00	14.00	18.00
21	14.00	5.00	11.00	14.00	11.00	14.00
22	2.00	14.00	16.00	16.00	13.00	18.00
23	4.00	10.00	3.00	11.00	11.00	14.00
24	11.00	6.00	6.00	18.00	14.00	18.00
25	11.00	2.00	5.00	16.00	13.00	5.00
26	12.00	11.00	4.00	6.00	8.00	5.00
27	8.00	13.00	10.00	9.00	13.00	12.00
28	6.00	13.00	5.00	1.00	10.00	16.00
29	5.00	4.00	8.00	16.00	7.00	19.00
30	5.00	5.00	4.00	9.00	12.00	24.00
Single Average Error		7.70	7.20	7.10	11.20	12.40
Average Error		7.30		13.10		

The robot 1 positioning error graph after reaching the given target points in individual experimental trials is presented in Figure 17.

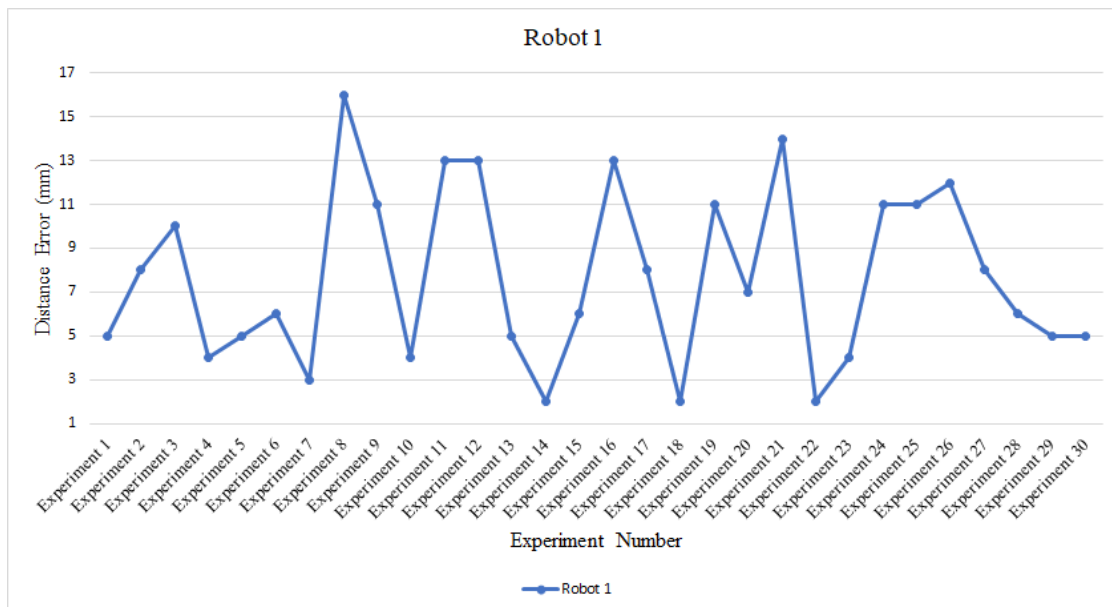


Figure 17: Positioning Error Rates of Robot 1.

The robot 2 positioning error graph after reaching the given target points in individual experimental trials is presented in

Figure 18.

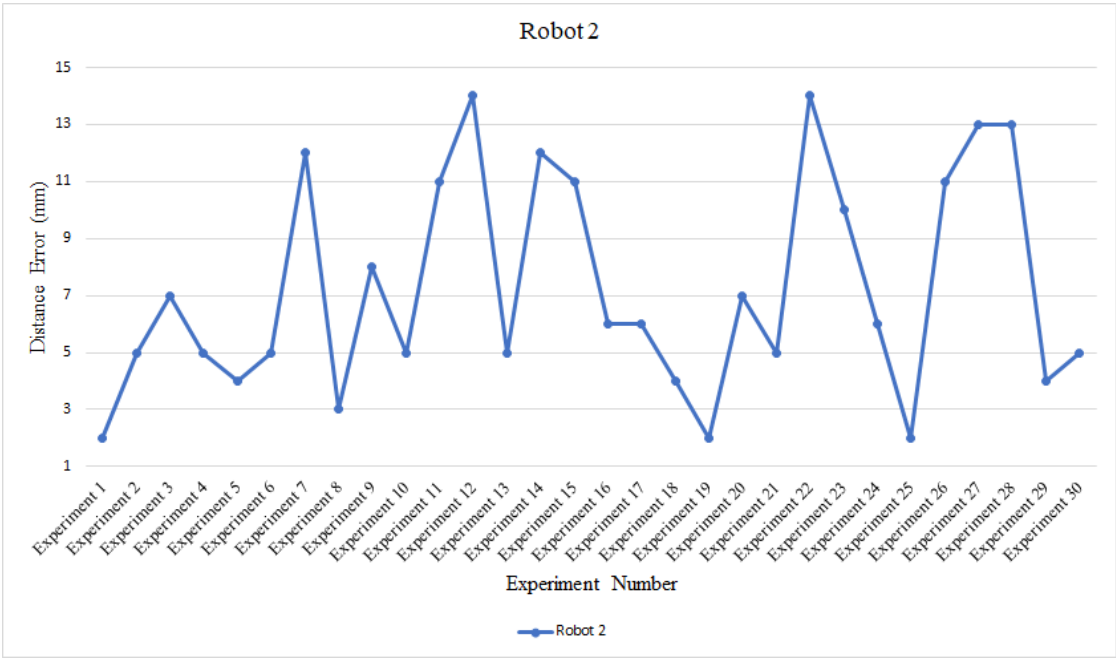


Figure 18: Positioning Error Rates of Robot 2.

The robot 3 positioning error graph after reaching the given target points in individual experimental trials is presented in Figure 19.

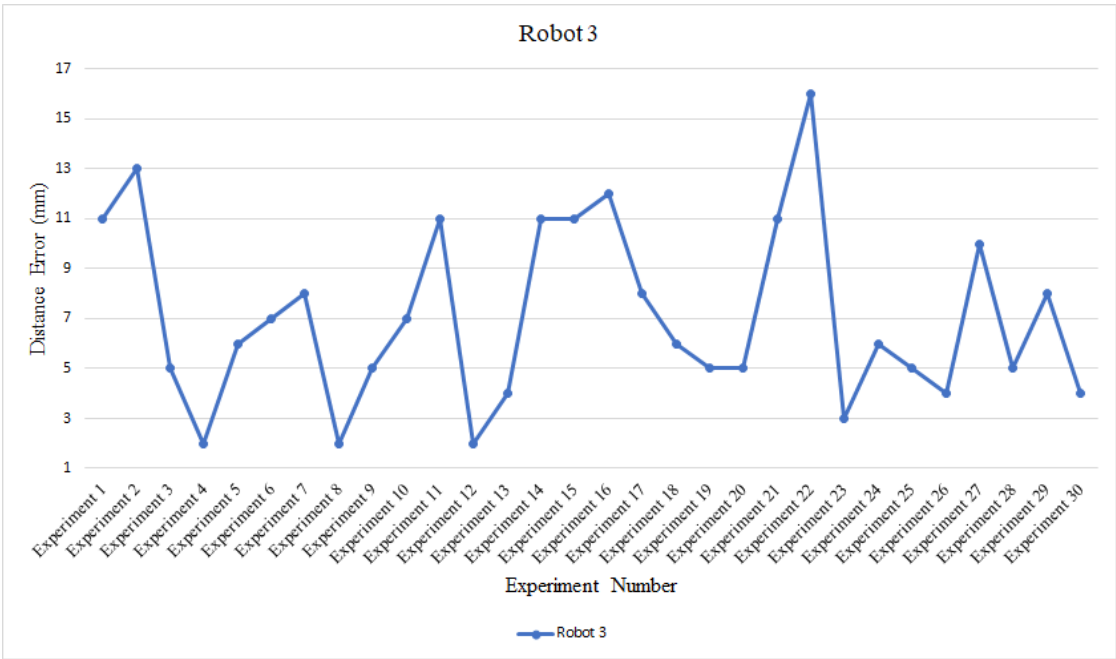


Figure 19: Positioning Error Rates of Robot 3.

The positioning error graph of all robots after reaching the given target points in multi-robot experimental trials is presented in Figure 20.

The experimental results indicate that the system positioning error averaged 7 mm in individual trials and 13 mm in

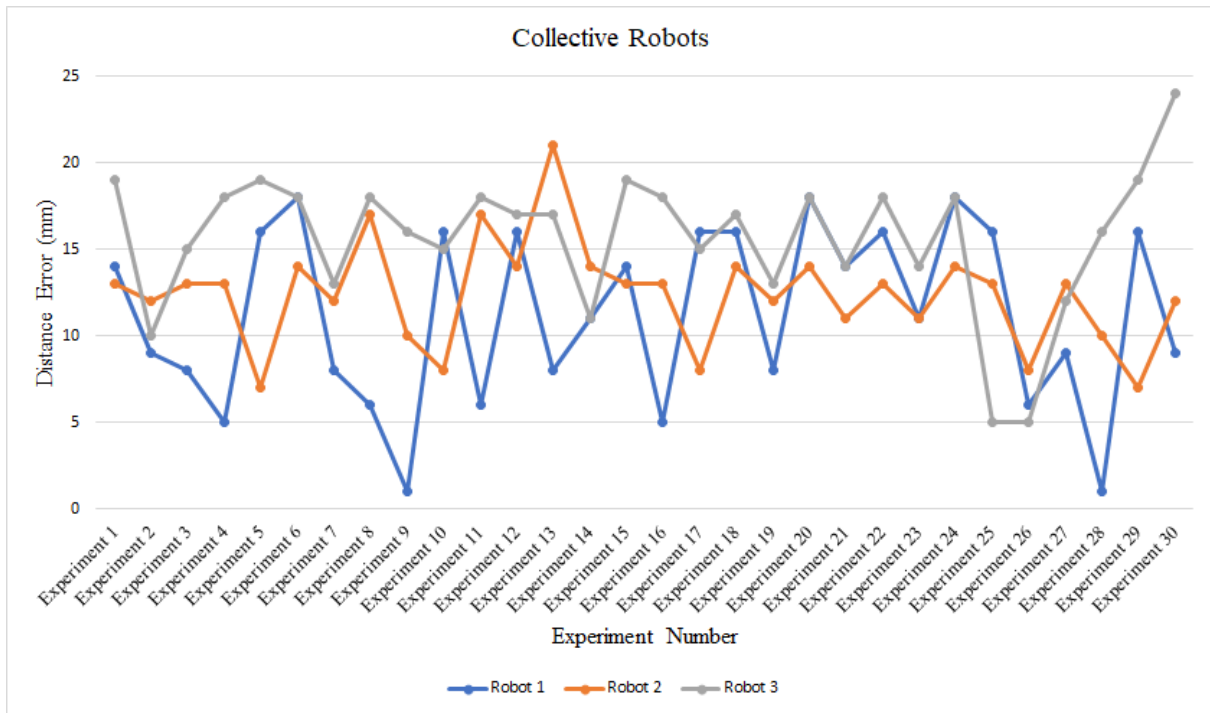


Figure 20: Positioning Error Rates of All Robots in the Multi-Robot Experimental Scenario.

multirobot trials. Considering that the average error rates in similar studies in the literature are generally above 100 mm, these findings demonstrate that our system exhibits significantly superior performance compared to its counterparts. In this context, we plan to continuously improve the components of the system, the testing environment, and the potential applications developed in this study. In particular, a multi-agent heterogeneous testing environment will be established to enable coordinated movement among robots. For this purpose, advanced robotic behaviors such as obstacle avoidance, consensus-based algorithms, and robotic formations will be implemented within the testing environment. Comprehensive tests will be conducted to improve the performance of the system and evaluate its effectiveness in various scenarios. Based on the data obtained, further enhancements will be made to the system. A comparison of similar studies in the literature with the system developed in this study is presented in Table 2.

Table 2: Comparison of Similar Studies in the Literature.

Method	Average Distance Error (mm)
Wi-Fi [18]	2310
Bluetooth [19]	15
IMU [20]	400
UWB [21]	40
Computer Vision [22]	30
Our Method	7

In conclusion, the exceptional performance and future development potential of the system developed in this study signify substantial advancements compared to existing research in the literature. Future investigations will concentrate on further optimizing the system and integrating it into new application domains. These efforts aim to empower robotic systems to function in a more intelligent, efficient, and harmonious manner.

Acknowledgments

This study includes the results of the Ph.D. thesis of the first author.

Authors' Contributions

Both authors contributed equally, read and approved the final manuscript.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] H. Sahib Hasan, M. Hussein, S. Mad Saad, and M. Azuwan Mat Dzahir, "An Overview of Local Positioning System: Technologies, Techniques and Applications," *International Journal of Engineering & Technology*, vol. 7, no. 3.25, p. 1, Aug. 2018. DOI: 10.14419/ijet.v7i3.25.17459.
- [2] A. Francis, A. Akinnusotu, and A. Adeyemi, "Global Positioning System and It's Wide Applications," *Continental J. Information Technology*, vol. 9, no. 1, pp. 22–32, Oct. 2015.
- [3] N. Syazwani C. J., A. Wahab, N. Sunar, H. Syed, K. Y. Wong, and Y. Aun, "Indoor Positioning System: A Review," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, Jan. 2022. DOI: 10.14569/ijacsa.2022.0130659.
- [4] S. Bastiaens, M. Alijani, W. Joseph, and D. Plets, "Visible Light Positioning as a Next-Generation Indoor Positioning Technology: A Tutorial," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 4, pp. 2867–2913, Fourth Quarter 2024. DOI: 10.1109/COMST.2024.3372153.
- [5] X. Zhu, W. Qu, T. Qiu, L. Zhao, M. Atiquzzaman, and D. O. Wu, "Indoor Intelligent Fingerprint-Based Localization: Principles, Approaches and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2634–2657, Fourth Quarter 2020. DOI: 10.1109/COMST.2020.3014304.
- [6] X. Feng, S. Liu, Q. Yuan, J. Xiao, and D. Zhao, "Research on wheel-legged robot based on LQR and ADRC," *Scientific Reports*, vol. 13, no. 1, p. 15122, Sep. 2023. DOI: 10.1038/s41598-023-41462-1.
- [7] O. Marques, *Practical Image and Video Processing Using MATLAB®*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2011. DOI: 10.1002/9781118093467.
- [8] A. Babinec, L. Jurišica, P. Hubinsky, and F. Duchon, "Visual Localization of Mobile Robot Using Artificial Markers," *Procedia Engineering*, vol. 96, pp. 1–9, 2014. DOI: 10.1016/j.proeng.2014.12.091.
- [9] G. Bradski and A. Kaehler, *Learning OpenCV*, O'Reilly Media, Inc., 2008.
- [10] M. Fiala, "ARTag, a fiducial marker system using digital techniques," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 590–596. DOI: 10.1109/CVPR.2005.74.
- [11] E. Tola, "Real-Time UAV Pose Estimation and Tracking Using FPGA Accelerated April Tag," RIT Digital Institutional Repository, 2021. URL: <https://repository.rit.edu/theses/10853>.
- [12] M. Nahangi, A. Heins, B. McCabe, and A. P. Schoellig, "Automated Localization of UAVs in GPS-Denied Indoor Construction Environments Using Fiducial Markers," *Proceedings of ISARC*, Jul. 2018. DOI: 10.22260/isarc2018/0012.
- [13] I. Ullah, D. Adhikari, H. Khan, M. Shahid Anwar, S. Ahmad, and X. Bai, "Mobile robot localization: Current challenges and future prospective," *Computer Science Review*, vol. 53, Aug. 2024. DOI: 10.1016/j.cosrev.2024.100651.
- [14] A. Sampathkrishna, "ArUco Marker-based localization and Node graph approach to mapping," *arXiv.org*, Aug. 2022. URL: <https://arxiv.org/abs/2208.09355>.
- [15] J. Kim, Y. Jeong, H. Lee, and H. Yun, "Marker-Based Structural Displacement Measurement Models with Camera Movement Error Correction Using Image Matching and Anomaly Detection," *Sensors*, vol. 20, no. 19, p. 5676, Oct. 2020. DOI: 10.3390/s20195676.
- [16] H. Bin, L. W. Zhen, and L. H. Feng, "The Kinematics Model of a Two-Wheeled Self-Balancing Autonomous Mobile Robot and Its Simulation," *Proceedings of the International Conference on Computer Engineering and Applications*, 2010, pp. 64–68. DOI: 10.1109/ICCEA.2010.169.
- [17] W. Burger and M. J. Burge, "Scale-Invariant Feature Transform (SIFT)", *Texts in Computer Science*, pp. 709–763, Jan. 2022. DOI: 10.1007/978-3-031-05744-1_25.
- [18] W. Cui, Q. Liu, L. Zhang, H. Wang, X. Lu, and J. Li, "A robust mobile robot indoor positioning system based on Wi-Fi," *International Journal of Advanced Robotic Systems*, vol. 17, no. 1, p. 172988141989666, Jan. 2020. DOI: 10.1177/1729881419896660.
- [19] D. R. Philips, E. Salami, H. Ramiah, and J. Kanesan, "Location Accuracy Optimization in Bluetooth Low Energy (BLE) 5.1-Based Indoor Positioning System (IPS)—A Machine Learning Approach," *IEEE Access*, vol. 11, pp. 140186–140201, 2023. DOI: 10.1109/ACCESS.2023.3338358.
- [20] F. Jiang, D. Caruso, A. Dhekne, Q. Qu, J. J. Engel, and J. Dong, "Robust Indoor Localization with Ranging-IMU Fusion," *arXiv*, Cornell University, Jan. 2023. DOI: 10.48550/arxiv.2309.08803.
- [21] M. Faeik, S. Junginger, T. Roddelkopf, and K. Thurow, "UWB-Based Real-Time Indoor Positioning Systems: A Comprehensive Review," *Applied Sciences*, vol. 14, no. 23, pp. 11005–11005, Nov. 2024. DOI: 10.3390/app142311005.
- [22] S. Roos-Hoefgeest, I. A. Garcia, and R. C. Gonzalez, "Mobile robot localization in industrial environments using a ring of cameras and ArUco markers," *Proceedings of IECON 2021-47th Annual Conference of the IEEE Industrial Electronics Society*, Toronto, ON, Canada, 2021, pp. 1–6. DOI: 10.1109/IECON48115.2021.9589442.