International Journal of Natural and Engineering Sciences
E-ISSN: 2146-0086 18(1): 12-31, 2024
Research Article

# A low-cost autonomous mobile robot localization using artificial landmarks

İbrahim Çayiroğlu[1], Ali Öztürk[2,*], Orhan Demirel[1,c]

[1]*Engineering Faculty, Department of Mechatronics, Karabük University, Karabük, Turkey*
[2]*Vocational School of Technical Sciences, Department of Machine and Metal Technologies, Muş Alparslan University, Muş, Turkey*
[3]*Industrial Vocational High School, Karabük, Turkey*

*Corresponding author: a.ozturk@alparslan.edu.tr

(Received: November 27, 2023 / Accepted: January 5, 2024)

**Abstract**

In this research paper, we present a robust, cost-effective, real-time localization method designed for both indoor and outdoor applications, leveraging a straightforward artificial landmark model. This method utilizes image data from artificial landmarks captured by a standard camera and a memory map. The robot computes its position by employing two landmarks prior to movement initiation and one landmark during motion. The use of a single landmark enhances the calculation speed. To enhance location accuracy, the results are subjected to filtering using a Kalman filter. Depending on the predefined constraints, the robot can autonomously or manually chart its course and navigate along it, achieving an average positional error of 4 cm.

**Keywords:** Mobile Robot, Localization, SLAM, Artificial Landmarks, Image Processing

## Introduction

Many tasks performed in everyday life involve operations conducted across extensive areas (such as surface cleaning, agricultural research, etc.) or point-to-point transportation applications. Employing robots in lieu of human labour for such continuous tasks presents significant advantages. This study introduces a mobile robot and localization system capable of detecting artificial land-marks and calculating position and orientation while adhering to predefined routes in both indoor and outdoor environments.

The design of autonomous mobile robots entails the consideration of several issues and challenges. The tasks essential for enabling a mobile robot to navigate autonomously can be categorized into four main components: (1) perceiving the environment, (2) constructing its representation of the environment, (3) determining its localization within the environment, and (4) devising and executing efficient navigation trajectories within this environment [1].

The issue of position estimation has garnered significant interest, leading to the development of numerous techniques for its resolution. Broadly, the approaches for ascertaining the location of mobile robots in the physical world can be categorized into two main groups: relative positioning and absolute positioning.

In the realm of relative positioning, methods including odometry (or dead reckoning) and inertial navigation [2] are commonly employed to determine the robot's position relative to an initial reference point with a high update rate [3]. Odometry remains a prevalent choice among internal sensors for real-time position estimation, primarily due to its practicality. However, information derived from encoders is inherently differential, and the presence of factors such as wheel slippage and quantization effects introduces inherent inaccuracies into these position estimates. As the robot progresses along its trajectory, these errors accumulate rapidly, leading to increasingly uncertain position estimates. Hence, frequent corrections from additional sensors, such as vision or range sensors, become imperative [4].

Conversely, absolute positioning relies on the detection and recognition of various environmental features to guide a mobile robot toward its destination and execute specific tasks. These environmental features are typically categorized into two types: feature-based, which includes elements such as active beacons [5],

artificial [6], and natural [7,8] landmarks, and map-based, which incorporates various geometrical models [9,10]. Vision-based localization commonly employs two types of landmarks for extracting distinct target images from background images: natural and artificial.

Natural landmarks are suitable for application in both indoor and outdoor environments. These objects or features exist within the environment and serve purposes other than aiding in robot navigation, such as corridors, floors, or doors [11]. They are chosen based on their distinctive geometrical or photometric attributes. Mobile robots analyze these attributes and leverage them to determine their own position and orientation [12,13]. Typically, natural land-marks are selected due to their unique characteristics. The mobile robot learns these characteristics, retains landmark features in its memory, and recognizes them as it moves, employing techniques such as Monte Carlo Localization [14], probabilistic methods [15] or based on learning domain-specific landmarks using a subspace projection method based on principal components analysis [16] .

Natural landmark is obtained based on the local features such as Scale Invariant Feature Transform (SIFT) [17, 18]. Based on the local features camera position can be localized, and the map of the environment can be reconstructed. For instance, Sa et al. [19] realized a Simultaneous Localization and Mapping (SLAM) application based on the natural landmarks obtained from the SIFT. However, this approach is highly sensitive to dynamic environments and is characterized by computationally intensive. Consequently, the extraction and utilization of natural landmarks for reliable real-time localization pose significant challenges. Methods utilizing natural landmarks are generally more versatile than those relying on artificial landmarks [20].

While approaches employing artificial landmarks may lack generality and necessitate some environmental modification, they exhibit notably precise and robust performance, even in intricate settings. Artificial landmarks are often designed with specific patterns or colors in mind to accommodate their detection algorithms [20]. Simultaneously, artificial landmark and active beacon methods are relatively inflexible. However, artificial landmarks prove to be highly effective tools for self-localization in indoor environments due to their simplicity [21]. They enhance the capability to locate landmarks and simplify the map-building process [3]. The advantages of land-marks characterized by distinct patterns include their ease of detection, identification, and tracking [22]. The artificial landmark-based approach imposes minimal computational overhead for image processing, rendering the extraction of landmarks from original image data more straightforward [23]. Nevertheless, a drawback lies in the requirement for the installation of numerous artificial landmarks [24].

Various techniques have been employed to enable a robot to determine its location within its environment. Prominent methods include triangulation [25-27], trilateration using landmarks [28-30], and a combination of these approaches [31-34].

In mobile robotics, triangulation involves the measurement of angles or bearings between landmarks to determine the robot's current position, while trilateration relies on distances to landmarks instead of bearing angles. Both trilateration and triangulation methods make use of landmarks for position estimation, and these land-marks can encompass various types, including infrared, ultrasonic, and radio frequency beacons, as well as geometric landmarks such as room corners and doors, among others.

In order to make the integration of mobile robots into daily operations feasible, it is essential to achieve a balance between costs and benefits. This paper primarily focuses on the utilization of artificial landmarks for position estimation in a cost-effective mobile robot.

**Related Works**

Throughout the years, a wide range of localization techniques have been proposed. This article delves into research exclusively focused on the artificial landmark method, highlighting the unique features that distinguish this study.

Martin et al. [35] performed pose estimation using a camera, capturing an octagonal landmark with nine dots positioned at its corners and in the center, as depicted in Figure 1.a. The study achieved relative errors of 1%

and pose errors of 2.6% within a range spanning 30 to 700 cm, utilizing a camera with a frame rate of 30 FPS (frame per seconds) with an image resolution of 640x480.

In Briggs et al.'s vision-based systems [36], an artificial landmark employs a self-similar intensity pattern in conjunction with a barcode for unique landmark identification, as illustrated in Figure 1.b. These landmarks are specifically designed for a navigation system that involves the construction of a topological map of landmark locations through environmental exploration, enabling navigation without relying on a global geometric map. Sensors are employed for land-mark detection and local obstacle avoidance. Camera images are captured at a resolution of 640x480 at 36 FPS and 320x240 at 81 FPS.

Kim et al. [24] endeavored to develop a practical solution for indoor mobile robot navigation, with a primary focus on achieving accurate absolute localization for enabling autonomous indoor navigation of a mobile robot, particularly for a ubiquitous home security service. The proposed system incorporates right-angled triangular landmarks as they offer a more straightforward means of determining the heading direction of mobile robots compared to other patterns. Each sensor node is equipped with three highly illuminating LEDs to create right-angled triangular landmarks, as illustrated in Figure 1.c. The PC camera used in the system possesses a resolution of 352x288 pixels and introduces an error of approximately ±5 cm within a range of 3 meters.

In the paper by Yoon [20], a straightforward color landmark model for self-localization is introduced, along with a rapid landmark detection and tracking algorithm that is founded on the proposed landmark model. This model employs a color landmark exhibiting symmetric and repetitive features, thereby demonstrating consistent color histogram characteristics even under certain geometric distortions, as depicted in Figure 1.d. The detection and tracking of this model are executed through a factored sampling technique, where-in color similarity is assessed using color histogram intersection. This color similarity is employed to update the color histogram model of the landmark for resilient tracking, particularly when subjected to variations in illumination.

Garcia and Romera [37] have developed an artificial landmark system designed for positioning in controlled enclosed environments, including pre-existing settings such as hospitals and homes. These landmarks take the form of standard A4 sheets of paper featuring four circles and two barcodes, as depicted in Figure 1.e. The circles serve as reference points for the system to compute both the distance and orientation to the landmark, while the right barcode corresponds to the landmark's unique identification number. The presence of the left barcode is intended to facilitate rapid landmark detection by the vision system, minimizing the occurrence of false positives during image processing.

Jang et al. [38] introduce a precise metric localization method employing a straightforward artificial landmark for indoor mobile robot navigation. The proposed landmark model is crafted with a three-dimensional, multi-colored structure, and the projective distortion of this structure encodes the robot's distance and heading relative to the landmark, as shown in Figure 1.f. Analysis of images captured by the camera, which offers a 360-degree field of view, enables the determination of the distance between the landmark and the camera. This determination relies on the shape distortion resulting from the angle of view and the alteration in shape size corresponding to the distance. The outcomes obtained indicate that the robot's self-positioning exhibits a maximum error of 7 cm.

Zitova et al.'s study [39] addresses the view-invariant recognition of circular landmarks employed in mobile robot navigation. The study introduces a recognition model based on affine moment invariants (AMIs). It explores the recognition capabilities of AMIs for this specific landmark shape under conditions involving additive random noise and/or varying viewing angles. The provided set of marks comprises patterns formed by two concentric circles with equal outer and differing inner radii, as depicted in Figure 1.g. The information intended for the robot can be encoded by using the ratio of the mark's inner and outer radii. This encoding method ensures that the acquired information remains independent of the robot's position concerning the mark. Marks created in this manner offer a single free parameter for data encoding, and their design facilitates the straightforward localization of these marks within complex environments.

In Arican's thesis [40], two vision-based robot localization algorithms, one utilizing artificial landmarks and the other relying on natural landmarks, are implemented. The first algorithm employs artificial landmarks

arranged around the robot. These landmarks are in the form of cylinders with strips designed for visual recognition of the landmark numbers (Figure 1.h). The camera system is mounted on the robot vehicle and captures the front scene of the robot.

Xu at al. [41] designated a mobile 3D printing construction robot. To overcome the insufficient localization of mobile robot in featureless scene an artificial landmark map is built in advance. Then, affected by the uncertainty of mapping, an accurate map is further constructed by a novel non-linear graph optimization method based on graphs. The localization accuracy of the mobile robot is improved by incorporating the reflectors attached with the high-reflective film for a large-scale ongoing scene.

Romero-Ramirez et al. [42] proposed a novel visual SLAM approach that efficiently combines keypoints and artificial markers, allowing for a substantial reduction in the computing time and memory required without noticeably degrading the tracking accuracy. In the first stage, the system creates a map of the environment using both keypoints and artificial markers, but once the map is created, the keypoints are removed and only the markers are kept.

In this research, akin to Arican's approach [40], a cylindrical-shaped landmark in orange color was developed. Differing from Arıcan's study, the strips on the landmark were employed for both distance measurement and landmark number determination. For indoor environments, the dimensions of the landmark were selected as 65 cm in length and 7 cm in thickness. With a landmark of this size, distance calculations can be made with less than a 1% error within an 8-meter range during movement.

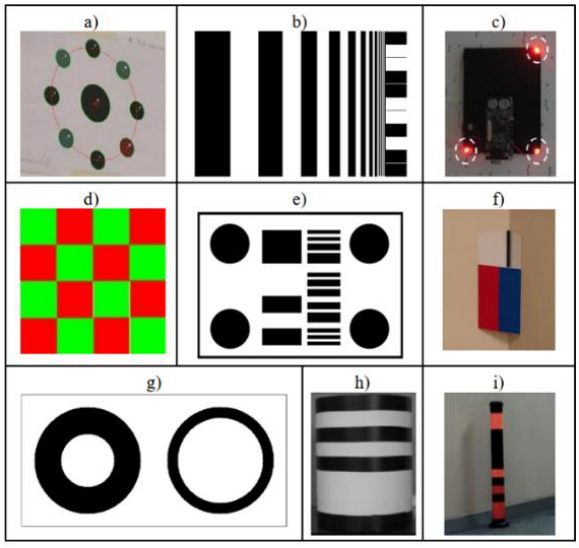



***Figure 1.*** *Examples of studies conducted using artificial landmarks. a) Nine dots (Martin, 2012) [35]. b) Self-similar landmark pattern with barcode (Briggs, 2000) [36]. c) Color landmark model (Yoon, 2001) [20]. d) Color landmark model (Yoon, 2001) [20]. e) Barcodes and four circles (Garcia, 2002) [37]. f) Two-color and three-dimensional shape (Jang, 2005) [38]. g) Two con-centric circles with equal outer and different inner radii (Zitova, 99) [39]. h) Barcode-similar landmark (Arican, 04) [40]. i) Bar-code-similar orange landmark (the study presented here).*

## Materials and Methods

### *Overall System*

The mobile robot has been designed to perform two primary tasks: 'Scanning Area' and 'Tracking Route.' The motion control system of the robot is configured as depicted in Figure 2.

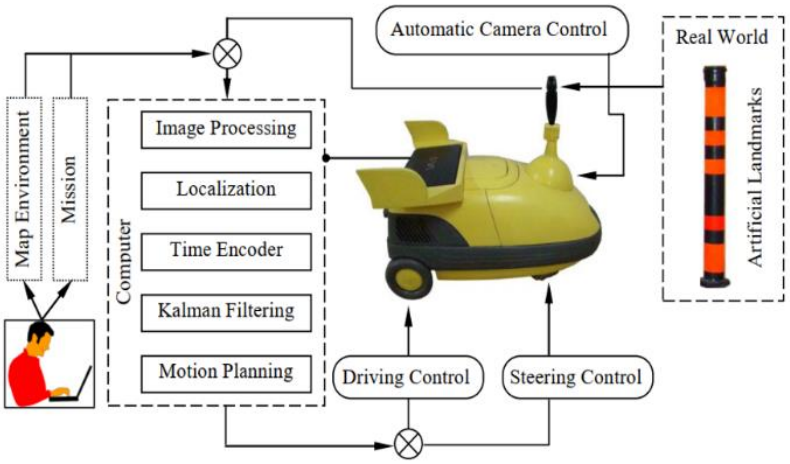



***Figure 2.*** *Block diagram of motion control system.*

The robot is propelled by 2 DC motors connected to the rear wheels, and steering is achieved through a servo motor installed on the front wheel. To control the robot, process acquired images, and execute navigation tasks, an Atom processor computer (1.88 MHz, 2GB RAM, 8 inches) is utilized. Additionally, a CCD camera (30fps, 5MP) with a $270^{\circ}$ continuous rotation capability is employed to capture images of the surrounding landmarks (Figure 3). Dedicated drive and microcontroller circuits were developed for motor control and communication with the computer, as illustrated in Figure 4. The computer on-board the robot processes images from the camera, generating movement data and transmitting it to the microcontroller circuit via USB. The microcontroller, in turn, sends movement and direction signals to the motor driver circuit for the DC motors and the servo motor.

***Figure 3.*** *Mobile robot platform for the experiment. It is controlled with 1.88 GHz atom microprocessor, 2 GB RAM, 8-inch netbook computer*

The computer processes images captured by the camera to generate motion information. It sends this information to the microcontroller circuit via Universal Serial Bus (USB). The microcontroller, in turn, transmits motion and direction signals to the robot's DC motor driver circuit and servo motor.

In the USB interface circuit, the Microchip-produced PIC18F4550 integrated circuit is used as the microcontroller, and the L298 integrated circuit serves as the motor driver. The codes required for programming the microcontroller in this circuit are written in the CCS (Custom Computer Service) PIC (Peripheral Interface Controller) C language. Special USB communication codes for the microcontroller used in the USB interface are added to the program as a library.

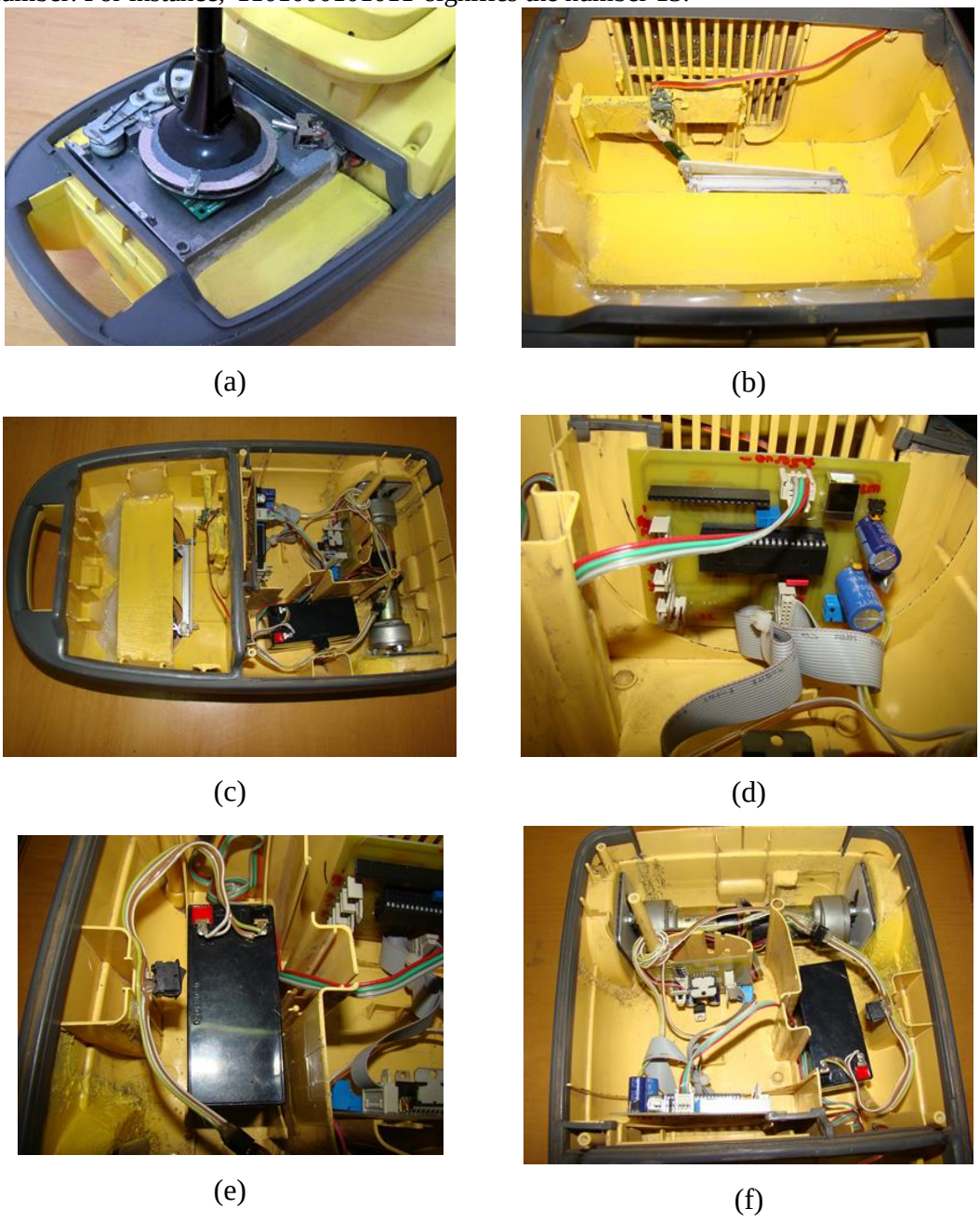
### *Artificial Landmark Based Vision Localization System*

To determine the robot's position, artificial landmarks were strategically positioned throughout the field site. These landmarks, situated in the environment, were identified by the camera, and based on the markings on the recognized landmark, the specific landmark and its dimensions were determined. The distance from the camera to the landmark was then calculated based on the gathered information.

A higher image resolution results in greater processing precision, but it also leads to an increase in processing time. Therefore, it was deemed sufficient to process an image of a landmark taken from a distance of 8 meters with a 1% margin of error. As a result, the acquired video frames were downscaled to images with a resolution of 400x300 pixels for processing.

The landmarks were designed in a cylindrical shape, with their surfaces alternately painted in black and orange colours. Additionally, these landmarks can be created as wall stickers if preferred. The length of each landmark was set at 65 cm, and 13 segments of 5 cm each were incorporated into them. These segments were painted in black and orange following a 4-bit binary numbering system to establish the desired landmark number. In the upper and lower sections, 10 cm segments were consistently painted orange to enable accurate reading and measurement of the landmark's length. The intermediate segments were painted either black or orange, based on the desired number, as depicted in Figure 5. It's important to note that only 15 landmarks can be uniquely numbered using a 4-bit binary system. If more landmarks are to be employed, the number of bits must be increased accordingly.

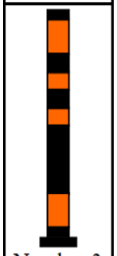
The orange and black regions on the landmark are translated into a digital code, with orange regions being denoted as '1' and black regions as '0'. The code derived from these binary numbers corresponds to the landmark number. For instance, '1101000101011' signifies the number 13.



(a) (b) (c) (d) (e) (f)

***Figure 4.*** *The internal electro-mechanical systems of the robot: a) Automatic rotation of the camera and potentiometer system. b) Servo motor system for direction control. c) General overview of the inside of the robot. d) USB interface circuit. e) Battery and connections. f) Reducer DC motors and driver circuit for motion.*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | Example |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **No** | Up | | | | | | | | | | | Down | | |
| **0** | 1 | 1 | 0 | **0** | 0 | **0** | 0 | **0** | 0 | **0** | 0 | 1 | 1 | |
| **1** | 1 | 1 | 0 | **1** | 0 | **0** | 0 | **0** | 0 | **0** | 0 | 1 | 1 | |
| **2** | 1 | 1 | 0 | **0** | 0 | **1** | 0 | **0** | 0 | **0** | 0 | 1 | 1 | |
| **3** | 1 | 1 | 0 | **1** | 0 | **1** | 0 | **0** | 0 | **0** | 0 | 1 | 1 | |
| **4** | 1 | 1 | 0 | **0** | 0 | **0** | 0 | **1** | 0 | **0** | 0 | 1 | 1 | |
| **5** | 1 | 1 | 0 | **1** | 0 | **0** | 0 | **1** | 0 | **0** | 0 | 1 | 1 | |
| **6** | 1 | 1 | 0 | **0** | 0 | **1** | 0 | **1** | 0 | **0** | 0 | 1 | 1 | |
| **7** | 1 | 1 | 0 | **1** | 0 | **1** | 0 | **1** | 0 | **0** | 0 | 1 | 1 | |
| **8** | 1 | 1 | 0 | **0** | 0 | **0** | 0 | **0** | 0 | **1** | 0 | 1 | 1 | |
| **9** | 1 | 1 | 0 | **1** | 0 | **0** | 0 | **0** | 0 | **1** | 0 | 1 | 1 | |
| **10** | 1 | 1 | 0 | **0** | 0 | **1** | 0 | **0** | 0 | **1** | 0 | 1 | 1 | |
| **11** | 1 | 1 | 0 | **1** | 0 | **1** | 0 | **0** | 0 | **1** | 0 | 1 | 1 | |
| **12** | 1 | 1 | 0 | **0** | 0 | **0** | 0 | **1** | 0 | **1** | 0 | 1 | 1 | |
| **13** | 1 | 1 | 0 | **1** | 0 | **0** | 0 | **1** | 0 | **1** | 0 | 1 | 1 | |
| **14** | 1 | 1 | 0 | **0** | 0 | **1** | 0 | **1** | 0 | **1** | 0 | 1 | 1 | |
| **15** | 1 | 1 | 0 | **1** | 0 | **1** | 0 | **1** | 0 | **1** | 0 | 1 | 1 | |


Number:3

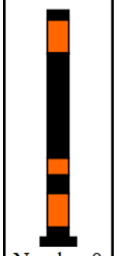

Number:8

***Figure 5.*** *Binary number system (4-bit) used for forming landmark shape.*

A rule-based algorithm was employed to detect the orange color in the landmark image. Depending on the lighting conditions within the robot's operational environment, a color range capable of recognizing the orange hue across a broad spectrum was established, as shown in Figure 6.a.

During the identification of the orange color, the following criteria were employed for the RGB (Red, Green, Blue) parameters:

a) The Red color value should fall within the range of 60-255.
b) The Green color value should range from 0-155.
c) The Blue color value should be within the range of 0-100.
d) The Red color value should consistently be 1.5 times greater than the Blue color value.

Pixels adhering to these criteria are transformed to the standard orange color (R=255, G=125, B=0), while the remaining pixels are converted to black. This results in an image composed of two colours, orange and black, as depicted in Figure 6.b. Subsequently, the number of pixels with orange color along the vertical axis (y-axis) is computed for each pixel along the horizontal axis (x-axis) and visualized as a graph, as shown in Figure 6.c. Landmark detection is performed within the regions along the x-axis where the number of values exceeds half of the maximum 'y' values on the graph.

A 1-pixel-wide strip is extracted from the central portion of the region on the graph where a landmark is detected, and distinctive features of the landmark are sought within this strip, as illustrated in Figure 6.d. A sequence of binary numbers is generated by encoding the black and orange regions on the strip as '1' and '0', respectively. This sequence is then compared with the reference values presented in Figure 5 to determine the landmark number. To validate the correctness of the landmarks, two consecutive and equivalent orange regions at the strip's edges are required. By measuring the exact length of the landmark in pixels, the distance from the camera to the landmark can be computed.

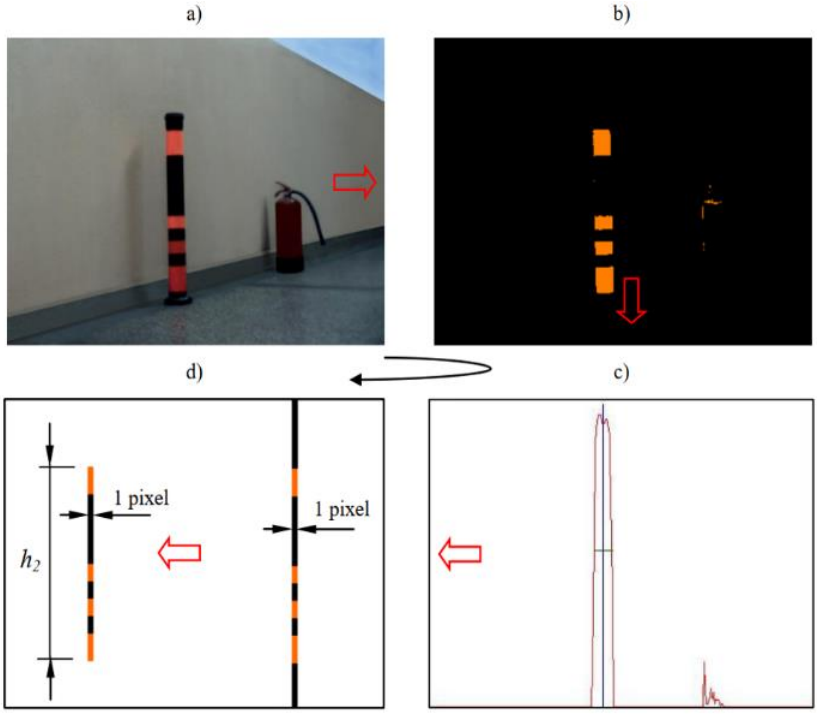



***Figure 6.*** *Landmark determination. a) An image frame captured by the camera. b) Thresholding for the orange color in the image. c) Generating the orange color graph. d) Extracting a 1-pixel-wide landmark image from the region where the landmark is potentially located in the graph.*

The shapes appear distorted due to the camera's rotation while reading the landmarks. This distortion is attributed to the camera's relatively low refresh rate. However, because the landmark's shape is intentionally designed to accommodate the horizontal rotation of the camera, this distortion does not pose any issues in distance measurement.

### *Localization of Mobile Robots*

To determine the robot's position in the 2D plane, it is necessary to capture images from at least three distinct points or locations. However, if the camera's viewing direction is already known, then capturing images from two points is adequate. Additionally, if information about the robot's previous positions, motion vector, and camera angle is available, a single image from a specific point can suffice for position calculation. In this study, location estimation is achieved using two landmarks while the robot is stationary, and with a single landmark while the robot is in motion.

The use of a smaller number of landmarks in location calculation can enhance calculation speed. The spatial distribution of the landmarks surrounding the robot has a more significant impact on location calculation accuracy than the quantity of landmarks. For instance, if the landmarks are concentrated at a distant point from the robot, it will lead to an increase in position error.

### *Measurement of the distance to Landmarks*

To calculate the robot's position, it is essential to measure the distance between the camera and the landmark. These measurements rely on the principle that 'distant objects appear smaller.' As objects shrink linearly with distance, a ratio can be established between the length of the landmark in the image and the actual distance. To achieve this, as depicted in Figure 7, the distance can be computed by inserting the necessary parameters into Equation 1. In this equation, $h_1$ represents the actual landmark length in cm, $h_2$ is the landmark length in the image in pixels, C denotes the experimentally determined constant for the camera (dimensionless), and r signifies the calculated distance between the camera and the landmark. For example, if a 65 cm-long landmark located 300 cm away from the camera is observed on the camera as 130 pixels, the constant C can be determined as 600. This constant remains consistent unless there are changes in the zoom or camera

settings. After obtaining the camera constant C, the distance can be computed using the landmark's pixel length in the images as follows;

$$\frac{h_2}{C} = \frac{h_1}{r} \Rightarrow r = \frac{Ch_1}{h_2}$$

***Equation 1***

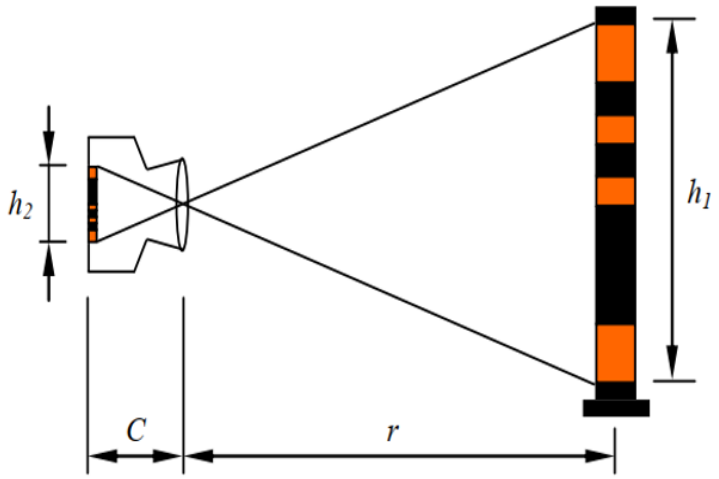



***Figure 7****. Distance measurement by using 'C-camera constant'.*

***Localization Using Two Landmark***

There are two solution sets for position calculations derived from two points. Determining which one of these solutions is correct can be accomplished using the camera's angle. To this end, a potentiometer was affixed beneath the camera to measure its rotational angles. Voltage variations resulting from the camera's rotation are digitized and processed by the microprocessor. As the obtained voltage changes form a linear graph based on the camera's rotation angle, angle values can be computed accordingly. The camera's rotation angle is set at $270^{\circ}$, which represents the maximum rotation angle for the potentiometer. The camera undergoes continuous rotation through a DC motor operating at reduced speed, and changes in direction are facilitated by altering the key polarities (see Figure 8).

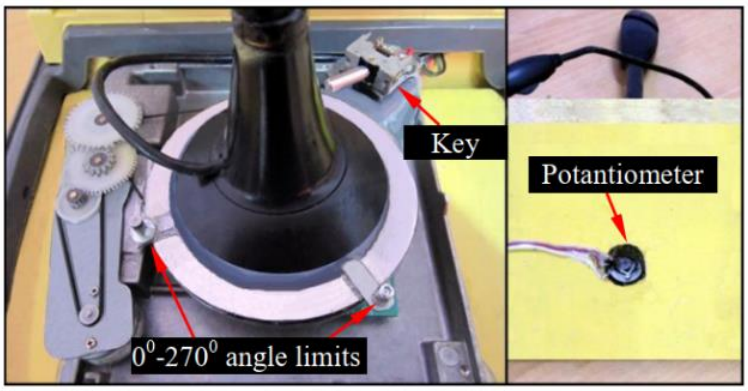



***Figure 8.*** *Camera rotation system and connections of the potentiometer*.

As depicted in Figure 9, when the robot calculates its position using the landmarks at points A and B, it results in two solution sets, denoted as $R_1$ and $R_2$, on the plane. Suppose, when the robot captures images of the landmarks at points A and B, the potentiometer records an angle of $150^{\circ}$ at point A and $60^{\circ}$ at point B. Consequently, when we plot the direction vectors $(v_1, v_2)$, the angle between them should be close to zero

for the correct location, whereas the angles between these vectors would be more substantial for the incorrect solution. In the example illustrated in Figure 9, the solution corresponds to point $R_1$.

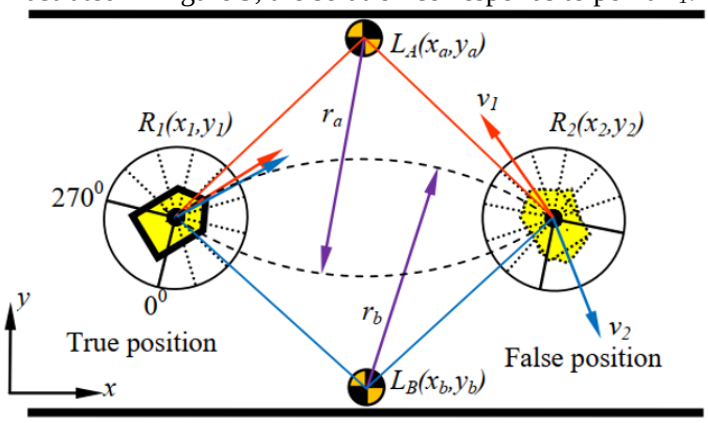



***Figure 9.*** *Location calculation by using two landmarks and camera angles.*

The potential robot locations, $R_1(x_1, y_1)$ and $R_2(x_2, y_2)$, are determined as the intersection points of two circles centered at landmarks $L_A(x_a, y_a)$ and $L_B(x_b, y_b)$, as follows;

$$(x_1 - x_a)^2 + (y_1 - y_a)^2 = {r_a}^2$$
$$(x_1 - x_b)^2 + (y_1 - y_b)^2 = {r_b}^2$$
$$(x_2 - x_a)^2 + (y_2 - y_a)^2 = {r_a}^2$$
$$(x_2 - x_b)^2 + (y_2 - y_b)^2 = {r_b}^2$$
$$d = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$$

***Equation 2***

The intersection points can be determined by solving the equations as follows;

$$A = \frac{x_a + x_b}{2} + \frac{(x_b - x_a)(r_a^2 - r_b^2)}{2d^2}$$
$$B = \frac{(y_b - y_a)}{2d^2}\sqrt{((r_a + r_b)^2 - d^2)(d^2 - (r_b - r_a)^2)}$$
$$x_1 = A + B$$
$$x_2 = A - B$$
$$C = \frac{x_a + x_b}{2} + \frac{(y_b - y_a)(r_a^2 - r_b^2)}{2d^2}$$
$$D = \frac{(y_b - y_a)}{2d^2}\sqrt{((r_a + r_b)^2 - d^2)(d^2 - (r_b - r_a)^2)}$$
$$y_1 = C - D$$
$$y_1 = C + D$$

***Equation 3***

The actual robot location, $R_1(x_1, y_1)$ and $R_2(x_2, y_2)$, can be accurately determined using the method described above, taking into account the angle obtained from the potentiometer. To enhance the precision of camera angle calculation, it is essential to incorporate the location that includes the landmark within the image into the calculations. By following this procedure with dedicated software, more precise angle calculations from the potentiometer can be achieved.

***Localization Using One Landmark***

Before the robot initiates its movement, it calculates the initial coordinates and direction based on two nearby landmarks, $L_A$and $L_B$ (Figure 10). Due to the minimal error in these calculations, it identifies an initial point, $R_1$, which may slightly deviate from the actual starting point, $R_0$. This point is considered as the initial location. Following the robot's movement, location calculations are conducted using a single landmark.

The robot continues its operation until it detects the presence of a third landmark ($L_C$). Utilizing its speed, direction, and elapsed time (Δt) from the time encoder, it calculates $R_2^1$. However, the distance $\left|R_2^2 - L_C\right|$ calculated by the image processing based on the third landmark may exhibit a slight discrepancy compared to the $\left|R_2^1 - L_C\right|$ distance due to inherent errors. Therefore, the initially calculated $R_2^1$. position is re-evaluated, and $R_2^2$. is considered as the new location. Further refinement of the $R_2^2$. point is achieved by applying a Kalman filter to obtain $R_2^3$., which aligns with the overall trajectory. Subsequent movement calculations are determined based on $R_2^3$ (Figure 10). While the robot is estimated to be at $R_2^3$, its actual position is $R_2^0$. The discrepancy between $R_2^0$ and $R_2^3$ points quantifies the error magnitude. If this error exceeds a specified tolerance (e.g., set at 20 cm), it indicates an unexpected situation. In such cases, the robot ceases its movement and re-establishes its position by employing two nearby landmarks."

Unexpected errors may arise from changes in the landmark positions, resulting in inaccurate distance measurements. Additionally, occlusion of a portion of the landmark strip by objects can render the landmark invalid or lead to the misreading of a different landmark number.

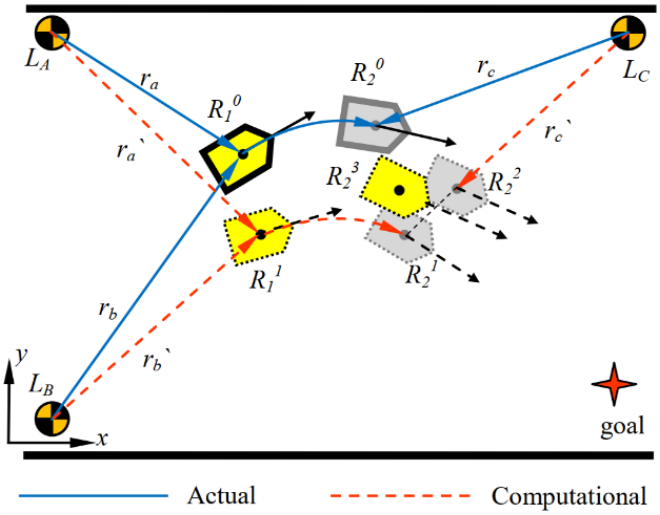



***Figure 10.*** *Position calculation from a single landmark during movement. $R_1^0$ represents the actual starting point, while $R_1^1$ signifies the calculated starting point obtained from $L_A$ and $L_B$. $R_2^0$ represents the real point at a time Δt later (as observed at the time of $L_C$). $R_2^1$ corresponds to the position calculated through time encoder calculations at Δt time later. $R_2^2$ denotes the correction point determined using $L_C$, and $R_2^3$ represents the estimated point adjusted with the Kalman filter.*

***Time Encoder Calculation***

While the robot transitions from point $R_1^1$ to point $R_2^1$ in Figure 10 (in actuality, from $R_1^0$ to $R_2^0$ points), it employs a temporal encoder in its location estimation. The movement trajectory is determined based on the robot's speed, which is calculated by considering the elapsed time and the angle of the front wheel.

In order to facilitate time encoder calculations, it is imperative to establish a reference point that traces a continuous and seamlessly flowing trajectory. Through simulations of the robot's movements, it becomes readily apparent that the most suitable point for consistently generating a smooth trajectory throughout various manoeuvres is the differential point 'D,' as visually depicted in Figure 11. As a result, time-dependent location calculations are executed with explicit reference to this differential point. The robot's motions and directional changes are executed in strict accordance with the steering system, as illustrated in Figure 12.

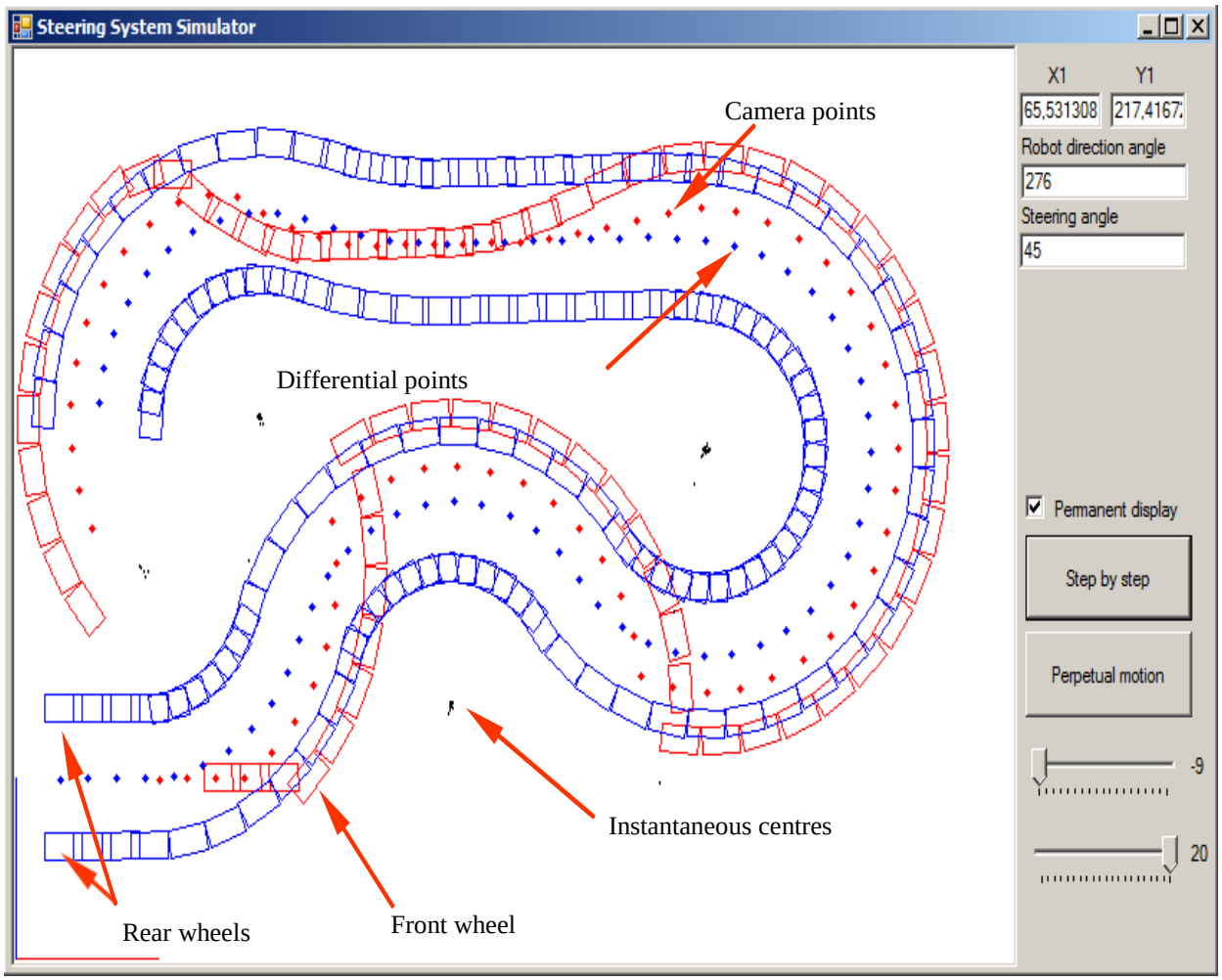



***Figure 11.*** *An analysis of the camera point, the differential point, and the positions and orientations of the wheels is conducted within the simulator. Given the linear trajectory that the differential point follows, time encoder calculations are performed with reference to this point.*

The distance between the instantaneous center of rotation (O) and the wheel center ($l_{OC}$) can be determined using the following formula based on geometric relationships;

$$l_{OC} = \tan(\beta - 270)\,(l_1 + l_2) - w/2$$

**Equation 4**

The coordinates of the differential point (D), positioned at the midpoint between the two rear wheels, are given as follows;

$$x_d = x_1 + l_2\cos(\alpha - 90)$$
$$y_d = y_1 + l_2\sin(\alpha - 90)$$

**Equation 5**

Let us determine the coordinates of the instantaneous center of rotation (O);

$$x_o = x_d + \left(l_{oc} + \frac{w}{2}\right)\cos(\alpha - 90)$$
$$y_o = y_d + \left(l_{oc} + \frac{w}{2}\right)\sin(\alpha - 90)$$

**Equation 6**

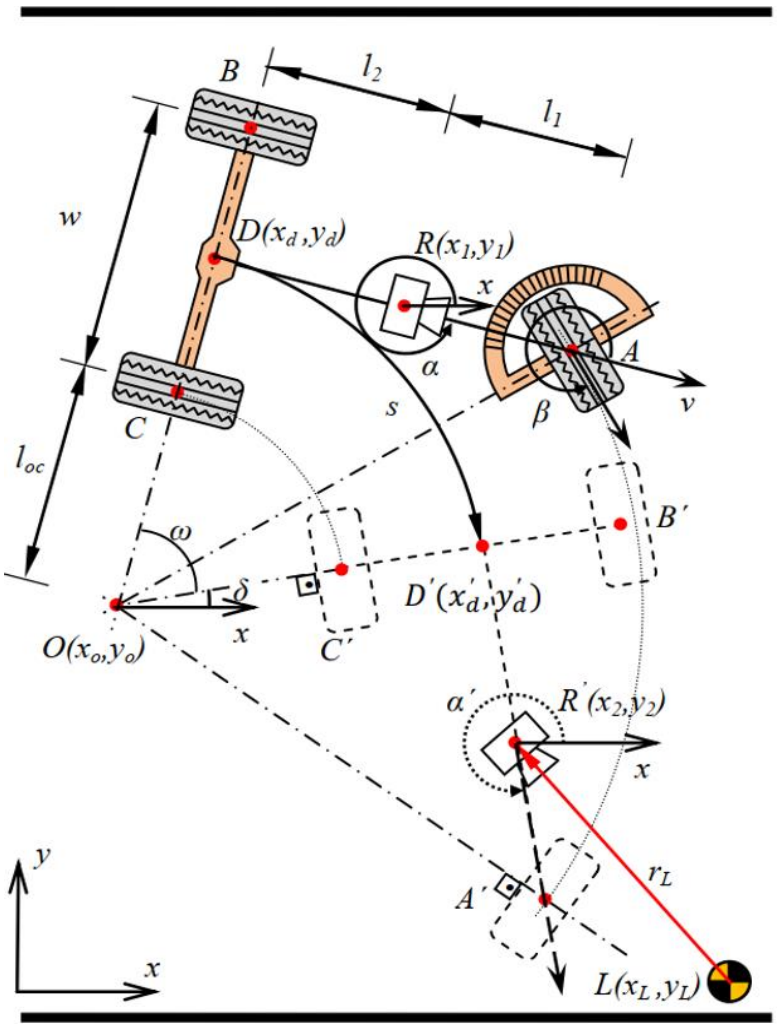



***Figure 12**. Time Encoder-Based Path Computation and the Ackermann Steering System*

The distance (s) traversed by the differential point during the robot's movement, the angle (w) that defines this displacement, and the angle (δ) formed by the line OD with the x-axis can be calculated as follows;

$$w = \frac{v\Delta t 360}{2\pi l_{OD}}$$
$$\delta = \alpha - (270 - w)$$

**Equation 7**

The new calculated position of the differential point ($D'$) will be as follows;

$$x_d' = x_o + \left(l_{oc} + \frac{w}{2}\right)\cos(\delta)$$
$$y_d' = y_o + \left(l_{oc} + \frac{w}{2}\right)\sin(\delta)$$

**Equation 8**

The new location of the robot ($R'$) can be calculated as follows;

$$x_2 = x'_d + l_2 \cos(270 + \delta)$$
$$y_2 = y'_d + l_2 \sin(270 + \delta)$$

**Equation 9**

### *Kalman Filtering*

In 1960, R.E. Kalman introduced an algorithm to address the linear filtering problem, which subsequently became known as the Kalman filter [43]. This algorithm comprises a set of mathematical equations and presents a recursive approach to solving the filtering problem. Later, an extension of this filter, known as the Extended Kalman Filter (EKF), was developed to handle non-linear systems. The filter employs statistical models to appropriately weigh each new measurement in relation to previous information to provide current estimates of system variables, such as position coordinates. Due to its exceptional performance, versatility, and ease of implementation, the Kalman filter has found widespread application in navigation systems [44].

The Kalman filter cycle, encompassing prediction and correction equations, is provided in Table 1 along with illustrative samples. As the robot moves, it adjusts itself based on the new coordinates estimated using the Kalman filter from the location measurements taken during its motion.

***Table 1.*** *Parameter and algorithm of Kalman filter*

| | **k:States (k={1,2,…})** | **k=1** | | **k=2** | |
|---|---|---|---|---|---|
| **Input Values** | $Z_k$ **: Measurement** | 130.24 (x-axis) | 174.51 (y-axis) | 138.84 | 177.12 |
| | **R: standard deviation of measurement (fixed)** | 0.6 (found experimentally) | 0.6 | 0.6 | 0.6 |
| | **Q: Noise value (fixed)** | 0.1 (found experimentally) | 0.1 | 0.1 | 0.1 |
| **Prediction** | $X_{k-1}$**:Previus estimation value** | 130.24 (initial estimation is equal to $Z_k$) | 174.51 | 130.24 | 174.51 |
| | $P_{k-1}$**:Previus error covariance value** | $P_{k-1}$=1 (found experimentally) | $P_{k-1} = P_k = 1$ | 0.352 | 0.352 |
| **Correction** | $K_k$**:Kalman Gain** | $K_k = \frac{P_{k-1} + Q}{P_{k-1} + Q + R} = 0.647$ | 0.647 | 0.43 | 0.43 |
| | $X_k$**:Corrected estimation** | $X_k = K_k Z_k + (1 - K_k) X_{k-1} = \mathbf{130.24}$ | 174.51 | 133.93 | 175.63 |
| | $P_k$**:Corrected error covariance value** | $P_k = (1 - K_k) P_{k-1} = 0.352$ | 0.352 | 0.201 | 0.201 |

### *Motion Planning*

The execution of a motion plan can be categorized into two primary methods. The first approach, known as 'Scanning Area,' entails the movement of the robot across the designated area to conduct a comprehensive survey within predefined boundaries. This method is typically selected for tasks such as cleaning and painting, where complete area coverage is imperative. The second method, 'Tracking Route,' is employed to guide the robot from one specified point to another along a predetermined path. This movement plan is particularly well-suited for applications involving the relocation of the robot from one location to another.

In the course of the 'Scanning Area' process, a pocket processing algorithm, reminiscent of milling operations, is utilized [45]. This involves creating enclosed loops by offsetting the drawn walls and subsequently interconnecting these loops. The robot initiates its movement from the outermost loop, commencing at the nearest perpendicular point, and proceeds in a counter clockwise direction to scan the nested loops. After completing each loop, the robot transitions to the next unprocessed loop. However, should the robot encounter a narrow passage that it cannot traverse, it becomes unable to access the other side of the area.

Conversely, when the 'Tracking Route' approach is chosen, it falls upon the user to input specific routes and destination coordinates into the program, as illustrated in Figure 13.

The robot follows the path by targeting the connection nodes, rather than adhering strictly to the path itself, during both the 'Scanning Area' and 'Tracking Route' processes. In doing so, it adjusts the angle of the front wheel to match the robot's orientation, allowing it to move smoothly towards the target point without unnecessary zigzagging. The robot's proficiency in route tracking is determined by its proximity to the nodes rather than the actual path, emphasizing the importance of accurately reaching these key reference points.

While the robot carries out its assigned tasks, it may encounter obstacles due to dynamic changes in the environment. A dynamic obstacle could be a person or an object that is not previously recognized or stored in its memory. To navigate around these obstacles, the robot is equipped with three proximity sensors located at the front of the robot, covering a 450-degree angle. When the robot detects an object, it instinctively initiates a leftward turn and then, while maintaining a safe distance of 30 cm from the obstacle, endeavours to re-join its intended route.

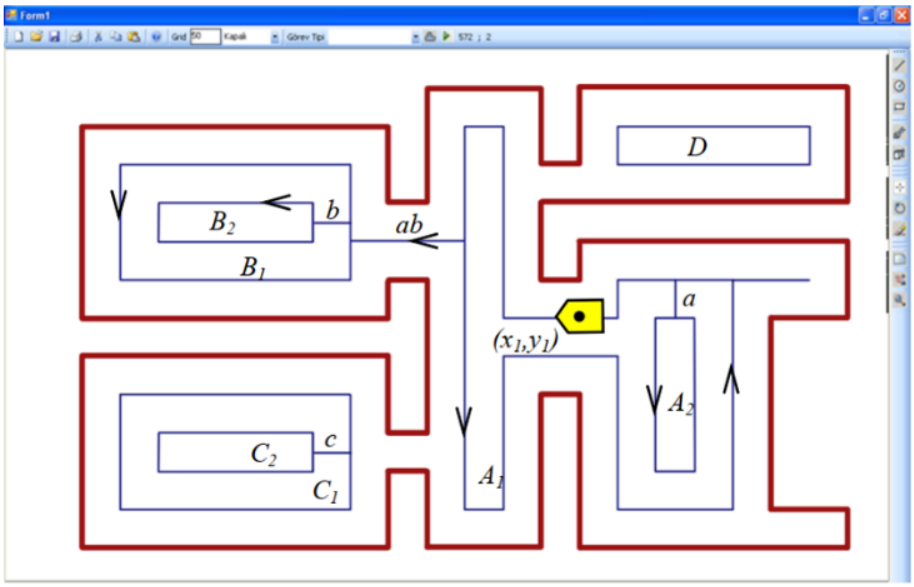



***Figure 13***. *Displaying the boundaries and routes on the program screen involves automatic route generation during the 'Scanning Area' process. When the robot commences movement from the point ($x_1$, $y_1$), it initially processes the $A_1$ and $A_2$ loops. Subsequently, it proceeds to handle the $B_1$ and $B_2$ loops. The robot, however, encounters an obstacle as it cannot process the $C_1$, $C_2$, and D loops due to the unavailability of a sufficiently wide gate. To navigate between the nested loops, it utilizes bridges denoted as 'a' and 'b,' while 'ab' connections are employed to link the groups of loops together.*

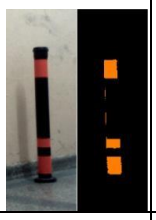
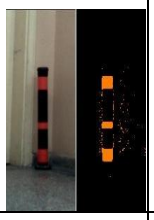
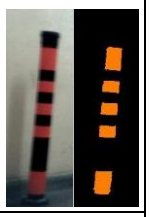
As the robot continuously logs every calculated location during its motion, any tasks that the robot cannot complete can be later reviewed and analysed graphically. Subsequently, a program can be devised to reattempt and successfully fulfil these pending tasks.

## Experimental Results

### *Distance Measuremnt Experiment*

Distance measurements from the images of six selected landmarks are provided in Table 2. These landmarks may appear distorted due to the camera's relatively slow image capture rate and the camera's rotation around its own axis. Nevertheless, the specific design of the landmarks and the algorithm employed effectively mitigate this issue. By incorporating features within the landmark design and post-processing the distorted images to remove a 1-pixel strip from the center, any curvature or distortion in the landmark's appearance is effectively resolved and is no longer a concern.

***Table 2.*** *Distance measuremnt made on the landmark images.*

| **Landmark Images** |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| **Landmark Number** | 8 | 4 | 12 | 7 | 7 | 7 |
| **Measuring Distance** | 254 cm | 339 cm | 724 cm | 240 cm | 238 cm | 193 cm |
| **Actual Distance** | 255 cm | 340 cm | 730 cm | 237 cm | 234 cm | 191 cm |
| **Error** | 0.39 % | 0.29 % | 0.82 % | 1.28 % | 1.7 % | 1.04 % |

In the experiments, where the distances to the landmarks ranged from 2 to 8 meters, a measurement error of approximately 1% was observed. This error is primarily attributed to specific conditions at the extreme ends of the measurement range. When measuring distances close to 2 meters, a portion of the landmark extends beyond the camera's field of view and thus is not captured in the image. Conversely, at distances exceeding 8 meters, the landmark image undergoes significant reduction in size, leading to an increase in measurement errors.

### *Route Tracking Experiment*

The algorithm's performance was evaluated through both real-world experiments, as shown in Figure 14, and simulations. Utilizing a simulator provided an opportunity to assess the algorithm's behavior in various scenarios and identify potential programming errors. In the simulation, real-time data inputs, including the robot's movements (driving, steering, camera rotation), noise, and distance measurements from landmarks, were integrated. The simulation excluded only certain factors, such as the friction between the wheels and the ground, as well as the effects caused by the Ackermann steering system during turns.

***Figure 14****. Testing environment.*

The performance of the localization and navigation system was assessed based on the results obtained from the simulation, as depicted in Figure 15. Notably, the simulation results closely matched the outcomes obtained from the real-world experiments, indicating the reliability and accuracy of the algorithm in both simulated and actual scenarios.

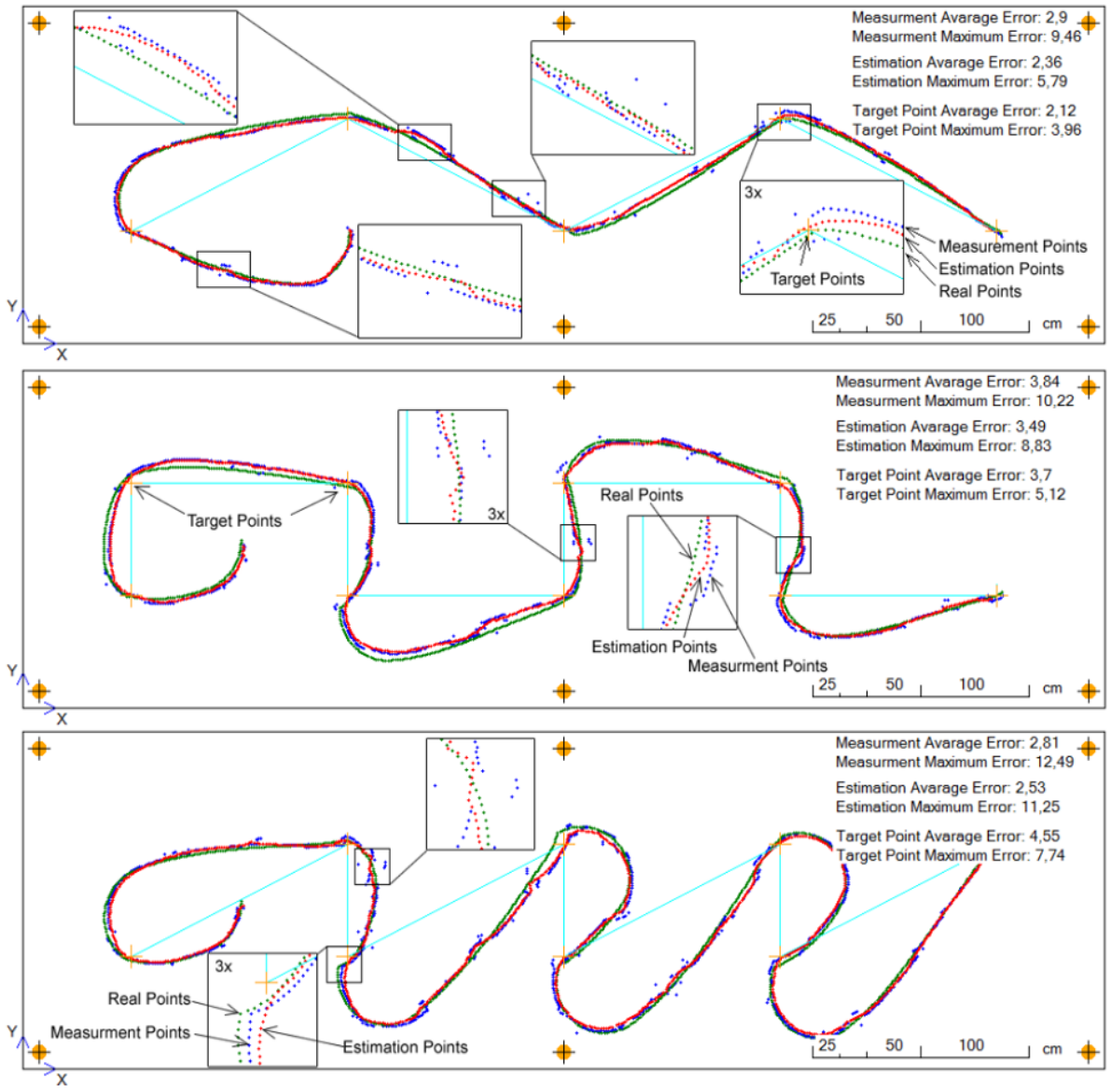



***Figure 15.*** *Showing 'real points', 'measurement points', 'estimated points' and 'target points', occurring during the movement of the robot in the simulation screen*

During the simulations, the calculation of the robot's position exhibited an average error of less than 4 cm when compared to both the experimental results and the estimations made using the Kalman filter. The incorporation of the Kalman filter contributed to an enhancement in measurement accuracy, resulting in an average improvement of 15% in the measurement results.

**Conclusion and Future Work**

In this research, an autonomously controlled mobile robot was employed for the calculation of its position and the determination of routes by processing artificial landmark images within its environment. Equipped with various attachments, the robot demonstrated its capabilities in performing a wide range of tasks, including interior tasks like cleaning, painting, and transportation, as well as outdoor tasks such as grass cutting, site clearance, and mine searching.

For outdoor applications, landmarks for the robot were established at strategic locations, including electric poles and building corners, where markers could be placed to aid in navigation and positioning.

From the research conducted on the developed robot, the following results and future prospects have been identified;

a) When utilizing 65 cm-long landmarks in the environment and capturing images at a resolution of 400x300 pixels for distances up to 8 meters, distance measurements can be achieved with less than a 1% error.
b) The successful performance of a simple camera suggests that this method can be effectively applied in various settings.
c) During the robot's movement, the use of single landmarks for position calculations enhances calculation speed.
d) Although there may be image shifts in low-speed camera capture, this does not pose any issues for the calculation algorithm and the shape of the landmarks.
e) To expedite calculations on the images, a lower resolution is favoured. In this study, a resolution of 400x300 pixels was employed, resulting in an average location calculation error of 4 cm. Higher resolutions tend to reduce the error.
f) The design of landmarks can be tailored to blend aesthetically with the environment, whether affixed to walls or used as decorative elements. Invisible paints can be used when employing infrared cameras.
g) This method can be adapted for open areas by increasing the size and diameter of the landmarks. For instance, by utilizing roadside electric poles as landmarks, it becomes feasible to employ this method for agricultural activities in open areas, or for operations on a campus or within a factory.
h) In a related study employing the same method, location calculations can be conducted in open environments, such as using a single landmark (e.g., an electric pole or factory chimney), with the camera continuously tracking the landmark for navigation.

**References**

[1] Talluri R., Aggarwal JK, Position estimation techniques for an autonomous mobile robot-A review, Handbook of Pattern Recognition and Computer Vision, World Scientific Publishing Company, 1998, pp. 765-796.

[2] Barshan CB, Durrant-Whyte HF, Inertial sensing for mobile robotics, IEEE Transactions on Robotics and Automation 11 (3) (1995) 328-342.

[3] Hu H, Gu D, Landmark-based navigation of industrial mobile robots, International Journal of Industry Robot 27 (6) (2000) 458–467.

[4] Borenstain J, The CLAPPER: a dual-drive mobile robot with internal correction of dead-reckoning errors, in: Proc. of IEEE International Conference on Robotics and Automation, San Diego, California, 1994, pp. 3085-3090. doi: 10.1177/0363546512458223

[5] Kleeman L, Optimal estimation of position and headzing for mobile robots using ultrasonic beacons and dead-reckoning, in:Proc. of IEEE Int. Conference on Robotics and Automation, Nice, France, 1992, pp 2582-2587.

[6] Durrant-Whyte HF, An autonomous guided vehicle for cargo handling applications, International Journal of Robotics Research 15 (5) (1996) 407-440.

[7] Leonard J, Durrant-Whyte HF, Mobile robot localization by tracking geometric beacons, IEEE Transactions on Robotics and Automation 7 (3) (1991) 376-382.

[8] Madsen CB, Andersen Cs, Optimal landmark selection for triangulation of robot position, Journal of Robotics and Autonomous Systems 23 (4) (1998) 277-292.

[9] Elfes A, Sonar-based real-world mapping and navigation, IEEE Journal of Robotics and Automation 3 (3) (1987) 249-265.

[10] Thrun S, Bucken A, Integrating grid-based and topological maps for mobile robot navigation, in: Proc. of the 13th National Conference on Artificial Intelligence, Portland, Oregon, 1996, pp. 128- 133.

[11] Borenstein J, Everett HR, Feng L, Where am I? Sensors and methods for mobile robot positioning, Research Report, The University of Michigan, 1996.

[12] Ayala V, Hayet JB, Lerasle F, Devy M, Visual localization of a mobile robot in indoor environments using planar landmarks, in: Proc. of IEEE International Conference on Intelligent Robots and Systems, Takamatsu, Japan, 2000, pp. 275-280.

[13] Mata M, Armingol JM, De La Escalera A, Salichs MA, A visual landmark recognition system for topological navigation of mobile robots, in: Proc. of 2001 IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001, pp. 1124 - 1129.

[14] Dellaert F, Burgard W, Fox D, Thrun S, Using the condensation algorithm for robust, vision-based mobile robot localization, in: Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, vol. 2, 1999, pp. 588-594.

[15] Murrieta-Cid R, Briot M, Vandapel N, Landmark identification and tracking in natural environment, in: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Victoria, BC, Canada, 1998, pp. 179–184.

[16]. Sim R, Dudek G, Mobile robot localization from learned landmarks, in: Proc. of IEEE/RSJ Conference on Intelligent Robots and Systems, Victoria, BC, Canada, 1998, pp. 1060-1065.

[17] Ozturk A, Cayiroglu I, (2022). Performance of Interpolated Histogram of Oriented Gradients on the Feature Calculation of SIFT. Advances in Electrical & Computer Engineering, 22(3).

[18] Ozturk A, Cayiroglu I, (2022). A real-time application of singular spectrum analysis to object tracking with sift. Engineering, Technology & Applied Science Research, 12(4), 8872-8877.

[19]. Se S, Lowe DG, Little J, Vision- based global localization and mapping for mobile robots, IEEE Transactions on Robotics 21 (3) (2005) 364- 375.

[20] Yoon KJ, Kweon IS, Artificial landmark tracking based on the color histogram, in: Proc. of International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, 2001, pp. 1918-1923.

[21] Jang G, Kim S, Lee W,Kweon I, Color landmark based self-localization for indoor mobile robots, in: Proc. of IEEE International Conference on Robotics and Automation, Washington, D.C., 2002, pp. 1037- 1042.

[22] Ahn SJ, Rauh W, Recknagel M, Circular coded landmark for optical 3D-measurement and robot vision, in: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyongju, Korea, 1999, pp. 1128-1133.

[23] Atiya S, Hager GD, Real-time vision-based robot localization, IEEE Transactions on Robotics and Automation, 9 (6) (1993) 785-800.

[24] Kim JG, An J, Lee KD, Localization of mobile robot based on fusion of artificial landmark and RF TDOA distance under indoor sensor network, Int. Journal of Advanced Robotic Systems, 8 (4) (2011) 203- 211.

[25] Deans M, Hebert M, Experimental comparison of techniques for localization and mapping using a bearing- only sensor, in: Proc. of ISER '00 Seventh International Symposium on Experimental Robotics, Honolulu, Hawaii, 2000.

[26] Sim R, Dudek G, Mobile robot localization from learned landmarks, in: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Victoria, B.C. Canada, 1998, pp. 1060–1065.

[27] Betke M, Gurvits L, Mobile robot localization using landmarks, IEEE Transactions on Robotics and Automation, 13 (2) (1997) 251-263.

[28] Goel P, Robust localization using relative and absolute position estimates, in: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999, pp. 1134-1140.

[29] Kantor GA, Singh A, Preliminary results in range-only localization and mapping, in: Proc. of IEEE

Conference on Robotics and Automation, Washington, D.C., 2002, pp. 1818-1823.

[30] Moerno L, Armingol JM, Garrido S,De La Escalera A, Salichs MA, A genetic algorithm for mobile robot localization using ultrasonic sensors, Journal of Intelligent and Robotic Systems, 34 (2) (2002) 135- 154.

[31] Tardos JD, Neira J, Newman PM, Leonard JJ, Robust mapping and localization in indoor environments using sonar data, International Journal of Robotics Research, 21 (4) (2002) 311-330.

[32] Tomatis N, Nourbakhsh I, Siegwart R, Combining topological and metric: A natural integration for simultaneous localization and map building, in: Proc. of the Fourth European Workshop on Advanced Mobile Robots, Lund, Sweden, 2001, pp. 19-21.

[33] Se S, Lowe D, Little J, Vision-based mobile robot localization and mapping using scale-invariant features, in: Proc. of IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001, pp. 2051- 2058.

[34] Carbonaro A, Zingaretti P, Landmark matching in a varying environment, in: Proc. of Euromicro Workshop on Advanced Mobile Robots, Brescia, Italy, 1997, pp. 147-153.

[35] Martin A, Adan A, 3D real-time positioning for autonomous navigation using a nine-point landmark, Pattern Recognition, 45 (2012) 578–595.

[36] Briggs AJ, Scharstein D, Braziunas D, Dima C, Wall P, Mobile robot navigation using self-similar landmarks, in: Proc. of IEEE International Conference on Robotics and Automation, 2000, pp. 1428–1434.

[37] Garcia JGC, Romera MM, Quintas MM, Ureria JU, Positioning and localization system for autonomous wheelchair, in: Proc. of 28th Annual Conference of the IEEE Industrial Electronics Society, Sevilla, Spain, 2002, pp. 1555–1560.

[38] Jang G, Kim S, Kim J, Kweon I, Metric localization using a single artificial landmark for indoor mobile robots, in: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Alberta, Canada, 2005, pp. 2857-2862.

[39] Zitova B, Flusser J, Landmark recognition using invariant features, Pattern Recognition Letters, 20 (1999) 541-547.

[40] Arican Z, Vision-based robot localization using artificial and natural landmarks, M.Sc. Thesis, Department of Electrical and Electronics Engineering, Middle East Technical University, Turkey, 2004.

[41] Xu Z, Guo S, Song T, Li Y, & Zeng L, (2022). Localization of Mobile Robot Aided for Large-Scale Construction Based on Optimized Artificial Landmark Map in Ongoing Scene. CMES-Computer Modeling in Engineering & Sciences, 130(3).

[42] Romero-Ramirez F J, Muñoz-Salinas R, Marín-Jiménez M J, Cazorla M, & Medina-Carnicer R, (2023). sSLAM: Speeded-Up Visual SLAM Mixing Artificial Markers and Temporary Keypoints. Sensors, 23(4), 2210.

[43] Kalman RE, A new approach to linear filtering and prediction Problems, Transactions of the ASME, Journal of Basic Engineering, 82 (1960) 35-45.

[44] Levy LJ, The Kalman Filter: Navigation's Integration Workhorse, GPS World, (1997) 65-71.

[45] Goktaş M, Dilipak H, Guldas A, Analytical aproach for offsetting of two dimensional complex profiles, in: Proc. of 5th International Advanced Technologies Symposium, Karabuk, Turkey, 2009, pp. 1322-1329.