

Article

Developing a Machine Learning Algorithm for Service Robots in Industrial Applications

Nizamettin Kulaç¹ and Mustafa Engin^{2,*} ¹ Mechatronics Engineering Department, EGE University, İzmir 35100, Turkey² Ege Higher Vocational School Electronics and Automation Department, EGE University, İzmir 35100, Turkey

* Correspondence: mustafa.engin@ege.edu.tr; Tel.: +90-533-4350859

Abstract: Robots, which have mostly been effective in areas such as industrial, agricultural, and production facilities, have started to take a place in the service sector, as their technologies have become lower in cost and more easily accessible. This situation has attracted the attention of companies and researchers and has accelerated studies on this subject. In this study, an algorithm was developed for the autonomous mobile robot to serve in industrial areas. In line with this study, it was ensured that the autonomous mobile robot mapped the working environment, determined the working station in this environment, and then carried out transport operations between these working stations in accordance with a given work order. After the mobile robot fulfilled the work order, it went into a waiting state until a new work order was received. For the mobile robot to save energy, it was ensured that it waited close to the point where the work order came in the most, by means of machine learning in the waiting position. The developed algorithms were designed using the NI LabVIEW environment and then simulated in the RobotinoSIM environment and physically tested using the Robotino autonomous mobile robot platform. The experimental results showed that mapping and location reporting using an RGB camera, odometry, and a QR code eliminated permanent location errors, and the robot completed 50 work orders with 100% accuracy.

Keywords: autonomous mobile robot; navigation; dynamic environments; vision-based simultaneous localization and mapping; QR code; machine learning; robot vision



Citation: Kulaç, N.; Engin, M. Developing a Machine Learning Algorithm for Service Robots in Industrial Applications. *Machines* **2023**, *11*, 421. <https://doi.org/10.3390/machines11040421>

Academic Editor: Dan Zhang

Received: 1 February 2023

Revised: 13 March 2023

Accepted: 23 March 2023

Published: 25 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, the usage areas of autonomous mobile robots have been increasing rapidly, and these robots affect a large part of human life depending on this increase. Robots, which have mostly been effective in areas such as industrial, agricultural, and production facilities, have started to take a place in the service sector, as their technologies have become lower in cost and more easily accessible. This situation has attracted the attention of companies and researchers, accelerating studies on this subject. In the future, autonomous robots will form a large part of our lives [1,2].

Technological developments that increase the level of digitalization in production facilities and warehouse automation have accelerated with the emergence of the concept of Industry 4.0 [3]. This process has made it necessary to manage industrial robots, mobile robotic systems, sensor networks, and all production tools in warehouses or factories, even products in warehouses, from a central server [4]. In this area, there are deficiencies, such as the localization of mobile robots, the manipulation of raw materials, intermediates, end products in the right places for appropriate tasks, the coordination of the entire system, the management of security, and alarm events that await solutions. It is impractical to fully automate production and distribution processes at all times. Interaction with people remains another exposed problem to be solved.

Our study proposes a solution algorithm that will enable mobile robots and other production tools in the factory and warehouse automation process to work in harmony. In

line with this study, it was ensured that the mobile robot mapped the working environment, determined the working points in this environment, and then carried out the transport operations between these working stations in accordance with the given work order. After the mobile robot fulfilled the work order, it went into a waiting state until a new work order was received. For energy-saving purposes, the mobile robot waited close to the point where the work order came in the most, by means of machine learning in the waiting position. Our research outputs contribute to solving the following issues:

- Machine-to-machine communication.
- Collision-free robot navigation in a dynamic environment.
- Software development algorithms that manage industrial automation.
- Our work validates the entire mapping, placement, and collision avoidance stages and the path planning stage with a vision-based sensor.
- In addition to navigation, it also uses the same sensor in object handling.

The remainder of the article is organized as follows. In Section 2, firstly, the robot used in the study is introduced, and then the NI LabVIEW software used to program the robot and the robot simulation software features are explained. Continuing this section, speech to text, quick response (QR) code reading, line tracking, and the vision-based SLAM algorithms developed in this study are explained. The mapping and placement algorithms developed in Section 3 were tested first in the simulation environment and then experimentally. In Section 4, the results obtained are presented and compared with the targets. Finally, the results of the study are summarized.

2. Related Work

The definition of a robot by Gonzalez-Aguirre et al. has been defined in three classes with a focus on the human–robot interaction [5]. According to the first definition, the robot completely replaces the human worker in a dangerous or polluted environment that is unsuitable for human work. According to the second definition, the robot works closely with a human to increase comfort or minimize discomfort. In the third definition, the robot works as a part of the human body. According to the first and third definitions, using the robot is not an alternative; it is a necessity, and its use is quite common [6]. In the second definition, there is no obligation, but the use of service robots has been increasing rapidly in recent years, especially after the pandemic. Service robots, as opposed to the robots used in the other two definitions, do not work in an area reserved for robots. For this reason, to work with people, they need to learn the environment and perform given tasks by using this information [7,8]. Service robots are defined as a type of autonomous robot that assists humans by performing repeatable tasks. A service robot that will work in a storage area is a good example of repetitive work. The storage of products in factory warehouses, their transport from one place to another with forklifts and similar transport vehicles, and their placement on shelves are usually done by human operators [9]. This process is repeated many times for each order received by companies. These processes require large numbers of employees, take significant time, and their effectiveness cannot be guaranteed [10].

The selection of material handling elements is very important in the manufacturing process. The selection of unsuitable material handling equipment will also disrupt the production process time. In addition, improper selection will result in increased material handling costs [11]. There are five types of material handling equipment commonly used in manufacturing. These are industrial forklifts, automated guided vehicles, conveyors, cranes, and hoists. There are five types of automated guided vehicles: tow trucks, pallet trucks, forklifts, light duty carriers, and assembly line vehicles [12]. Industrial automation is a new but rapidly advancing technology. There is no single solution for automation. There are diverse options. Within robotics, special attention is paid to mobile robots, as they can navigate their environment and are not fixed in a single physical location. The tasks they can accomplish are endless, and the possibilities for their contribution to our lives are innumerable [13].

In study by Tatang Mulyana et al., Festo's Robotino robot was assigned to the task of transporting materials to the right places as material handling equipment [14]. During transportation, grippers were used for Robotino. In the study, materials in yellow, blue, and red colors come from three different conveyors. The task of the service robot is to take the incoming different-colored materials to three different production stations according to their colors. In this study, inductive lines are used as a guide for the navigation system. Robotino can transport materials along predetermined lines. At the end of the study, the color of Robotino materials was determined by means of a webcam. After determining the color, it activated the gripper and grasped the material. After making sure that it was in front of the production station with the distance sensors, it left the material it brought to the production station. Robotino checked whether there was material in all conveyors; if there was material in the conveyor, it continued these processes according to its color.

Another study investigated robots providing home security. The project addresses the issue of anti-theft using modern methods, such as an autonomous mobile security robot. The main goal of the research is the development of autonomous navigation techniques that allow the robot to randomly follow a complex route and detect suspicious situations using onboard image processing. Robotino was used for this study. The mapping program is a safe tool to show navigation efficiency and the path of previous navigations. In the project, researchers used the odometry technique for mapping. With this mechanism, all detections were motion based, and the robot needed it as an indoor security robot [15].

Anıl Akkız worked on the design and control of a mobile autonomous library robot [16]. There is a constant exchange of books in university libraries. Students or teachers need to find the code of a book in the library on a website, find the shelf containing the code after learning the code, and finally find the book on the shelf to borrow. With the robot designed in the study, it is intended to help the user access the book they are looking for in a comfortable way and to save time for the user by facilitating this access. In addition, it aims to place the books on shelves by eliminating human errors. This autonomous library robot design consists of two main parts. One of them is a mobile platform that enables the movement of the robot. The other is a robotic arm that determines the detection and retrieval of books. In the study, the need for service robots designed for the same or similar purposes, analysis, project management, and how to produce robots for specific tasks are shown. The robot produced as a result of the study was able to take itself to the target position, avoid dynamic objects, detect the desired book, and deliver it to the end user.

Rothomphiwat et al. developed a factory model controlled by robots in their study [17]. For the study, an autonomous drone, an autonomous mobile manipulator robot, and an autonomous dual-armed robot were used. The researchers set up a 6.5 m × 8.5 m factory environment for these collaborative robots to work. In the study, robots are designed to cooperate to move an object from one place to a shelf or storage area. They assumed an area that the mobile robot could not reach. The autonomous drone used was used to move the object to a place within reach of the mobile robot. This platform has been demonstrated through message queuing telemetry transport (MQTT), a common data communication protocol used for the Internet of Things (IoT) and robot operating systems (ROS). From advanced collaborative robots working in a factory environment, the drone started flying to deliver an object to the target location and then returned to the home position. The mobile manipulator sent a message via MQTT to activate the robot to take the object onto a conveyor while returning to the home position. After the mobile robot placed the object on the conveyor, it sent a message via MQTT to activate the conveyor. The conveyor moved the object to a position where the double-armed robot could detect and grasp it. The double-armed robot picked up the object and carried it to the human worker. Finally, the human worker delivered the object to the double-armed robot, and the double-armed robot placed the object on the shelf. This experiment was repeated 5 times with a 100% success rate. The platform has also demonstrated the potential of a new manufacturing process that incorporates adaptive and collaborative intelligence to increase the efficiency of mass customization.

Chan et al. proposed using a light detection and ranging (LIDAR) sensor to scan the interiors of commercial and industrial buildings in real-time 3D and, using the obtained data, a simultaneous localization and mapping (SLAM) algorithm for mobile robots [18]. Hanagi et al. suggested using the D* algorithm for simultaneous mapping and navigation using cheaper ultrasonic or infrared rangefinder sensors instead of using expensive LIDAR [19]. In the case of using the D* algorithm, using absolute audiometry, the simultaneous mapping and navigation problem in industrial environments can be approximately solved without using external sensors.

Tsintotas et al. proposes a fast and accurate indexing technique for location recognition in SLAM when action vision is used [20]. They saved time performing real-time vision loop-closure by using the Burkhard Keller tree instead of the bag-of-tracked-words model. With the experimental results, they have shown that more costly approaches can perform high-accuracy location identification.

Bai et al. reviewed the development of vision-based navigation and guidance applied in agricultural autonomous vehicles and robots [21]. They stated that the success of the use of vision-based SLAM in agricultural robots is of great importance for the selection of sensors suitable for the working environment. In the environmental data processing and navigation line extraction section, after summarizing the filtering-based data calculation method, segmentation-based data calculation method, and line detection-based data calculation methods, the authors stated that the combined use of these methods, depending on the product and environment, yields more successful results. The authors stated that the still unresolved problems in agricultural robots in the VSLAM method after combined use were light conditions, dust, different growth stages of plants, shading, and leaf overlap.

Narazaki et al. developed an approach for vision-based autonomous unmanned aerial vehicle (UAV) navigation planning for the rapid post-earthquake inspection of reinforced concrete railway viaducts [22]. The approach mimics the way human inspectors perform the task: the system does not require a complete 3D model of the environment and instead uses key features of the target structure as prior knowledge. The system parses sparse point cloud data using frame-by-frame semantic partitioning results online and defines the approximate scale using frame-by-frame depth estimation results. Then, rectangles are placed at the points labeled “Column”, and columns are determined by finding near-parallel or near-perpendicular pairs. In addition to columns detected in this way, the approach attempts to find a grid pattern of column positions based on which detection results are improved, and information about missing columns is extracted. With this approach, the UAV can initiate the inspection task before obtaining full information about the target structure. The map can be progressively developed as the UAV acquires more images during inspection. The ability of the developed approach to detect columns robustly and reliably was evaluated using synthetic data. The results showed that all eight columns of target viaducts were detected with a centimeter accuracy across all 100 trials. Finally, the entire approach was demonstrated in a synthetic environment, demonstrating the significant potential of efficiently and rapidly collecting high-quality images for post-earthquake structural investigation.

Adamkiewicz et al. proposed a trajectory planner and exposure filter that allow robots to take advantage of neural brightness field (NeRF) representation for collision-free navigation [23]. The proposed method aims to further integrate perception and control in an active planning style, both by inducing trajectories to orient the camera in directions with greater gradient information and by using uncertainty measures calculated by the state estimator to reduce the risk of collisions.

3. Materials and Methods

3.1. Mobile Robot

Festo Robotino is a mobile robot with three independent wheels. This means that the controllable degrees of freedom are equal to the total degrees of freedom. By means of these three wheels, the robot can turn around itself and go in all directions. Robotino

uses an omnidirectional drive to move. The omnidrive is powered by 3 GR24x40 20W-12V electric motors with 32:1 reducer and incremental encoder on each motor. It has a bumper sensor around it that allows it to stand against impact. There are 9 infrared distance sensors placed at 40-degree intervals to measure the distance between it and the objects around it. There is an RGB camera on it that can be used for image processing. The robot can be controlled by a computer system via Wi-Fi and USB cable connection [24]. This system allows movements to be executed in all directions and is also self-rotating, so it is useful for designing and testing robots, motion, autonomous navigation frameworks, line followers, and obstacle avoidance. Additionally, the Robotino features a 5 cm-wide gripper with an extra 5 mm opening range and very limited gripping power, which requires a tray-type base to hold objects. This base has a connected limit switch to detect when the object has reached the holding position to be grasped and transported. The Robotino robot can be programmed with the graphical Robotino[®] View [25], MATLAB Simulink [26], and NI LabVIEW [27] software packages. In addition to these, the programs written can be simulated with Robotino[®] SIM, with limited capabilities.

The Robotino mechanism and its end view are shown in Figure 1. The omnidirectional drive consists of three Mecanum wheels, each individually controllable. These wheels are arranged at 120° angles. While creating the dynamic model, it is assumed that there is no slip between the wheel and the motion surface. In addition, the wheel contact friction forces in the direction perpendicular to the pulling force are ignored. Two coordinate frames are used in the modeling: the earth coordinate frame {W} fixed to the ground and the moving coordinate frame {M} fixed to the robot's center of gravity. The world coordinate frame q definition is explained in Equation (1).

$$q = [x \ y \ \theta]^T \quad (1)$$

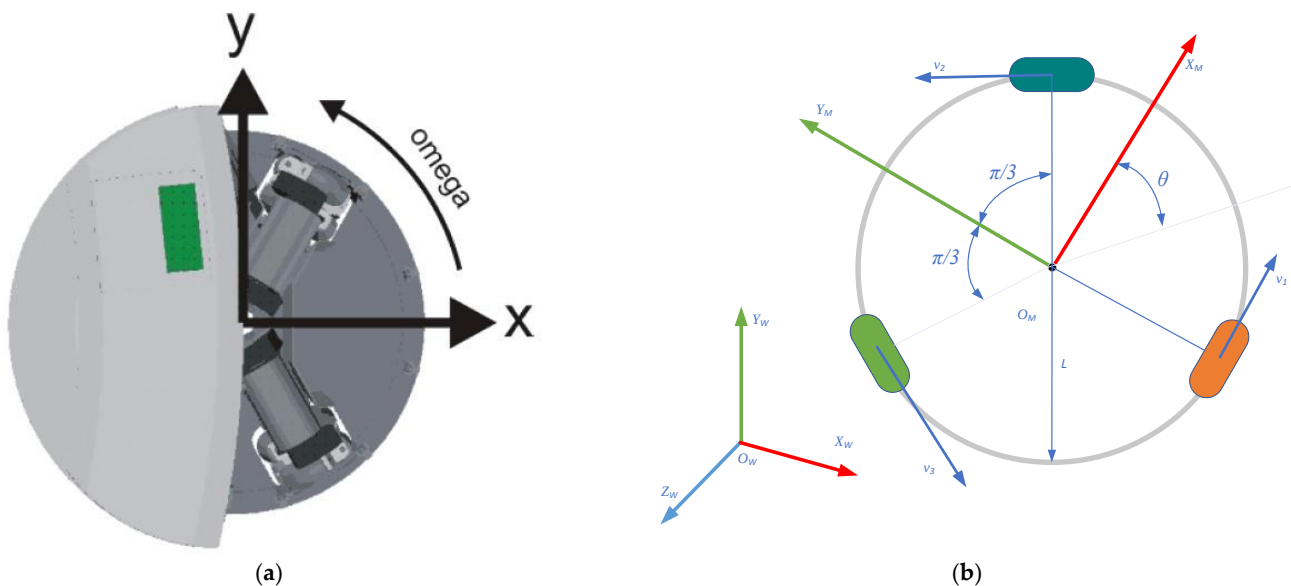


Figure 1. (a) Robotino mechanism and (b) world and mobile coordinate frames.

The moving coordinate frame V_M can be expressed with translational velocities and angular velocities as given Equation (2).

$$V_M = [V_x \ V_y \ \dot{\theta}]^T \quad (2)$$

V_x and V_y denote the translational velocities in the x - and y -axis directions, respectively, and $\dot{\theta}$ denotes the angular velocity around the z -axis. While the robot is in motion, it is

important to determine the coordinates of the robot according to the world frame [28]. For this, the moving frame world frame transforming matrix is defined in Equation (3).

$${}^W_M R = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

The speed of the robot according to the world frame is calculated using Equation (4).

$$\dot{q} = {}^W_M R V_M \quad (4)$$

According to the moving frame, the dynamic parameters of the robot F_x , F_y can be defined in Equations (5) and (6), respectively. The other dynamic parameter M_1 , moment of force around the axis of the robot's gravity center, is calculated from Equation (7).

$$m(\dot{V}_x - V_y \dot{\theta}) = F_x \quad (5)$$

$$m(\dot{V}_y - V_x \dot{\theta}) = F_y \quad (6)$$

$$I_v \ddot{\theta} = M_1 \quad (7)$$

The robot connects to the ground with the planar joint providing two translator motions p_x and p_y along the X_W and Y_W axis and rotation θ about the Z_W axis [29]. Inputs for the planar joint are given by

$$p_x = \int V_x dt = - \int v_1 \cos \left(\theta + \frac{\pi}{3} \right) dt + \int v_2 \cos(\theta) dt - \int v_3 \cos \left(\theta - \frac{\pi}{3} \right) dt \quad (8)$$

$$p_y = \int V_y dt = - \int v_1 \sin \left(\theta + \frac{\pi}{3} \right) dt + \int v_2 \sin(\theta) dt - \int v_3 \sin \left(\theta - \frac{\pi}{3} \right) dt \quad (9)$$

$$\theta = \int \frac{(w_1 + w_2 + w_3)D}{3L} dt \quad (10)$$

where v_1 , v_2 , and v_3 are linear velocities, and w_1 , w_2 , and w_3 are angular velocities of each wheel. R is the wheel radius. L is the radius of mobile robot platform.

3.2. NI LabVIEW

Programmers develop software applications every day to increase efficiency and productivity. LabVIEW, a programming language, is a powerful tool that can be used to achieve these goals. The name LabVIEW stands for Laboratory Virtual Instrument Engineering Workbench. It is a graphics-based programming language developed by National Instruments (NI). The LabVIEW language can be defined as a graphically structured data flow language. Graphical programming means a 2D diagram of blocks and interconnected wires rather than text. These blocks, called structure nodes, are used to represent loops and conditional states. Structure nodes have a diagram within blocks, and this diagram controls whether and when the internal diagram is executed. The inner diagram acts with the same semantics as the diagram outside the block. Structure nodes can be nested as in text-based scripting languages [30].

3.3. LabVIEW Text-to-Speech Synthesizer

A text-to-speech synthesizer generates speech against corresponding text inputs. It has been used to provide easier means of communication and to increase the access of visually impaired people to textual information [31]. Researchers work on adding more languages to the text-to-speech synthesizers and improving the sound quality. There are several ways to translate the text to speech in LabVIEW [32]. In this study, we used the

.NET callback event resource in LabVIEW to receive messages from the Speech Synthesizer assembly to perform updates during speech. The blocks given in Figure 2 are used for the program that converts text to speech.

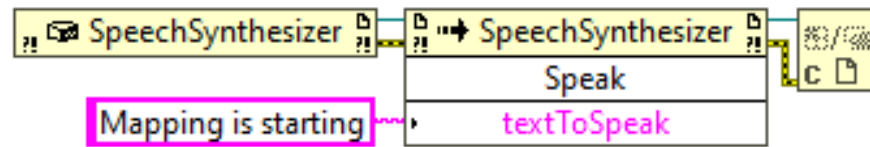


Figure 2. Algorithm that audibly announces the start of mapping.

3.4. RobotinoSIM

Robotino[®] SIM is a simulation environment developed to experiment with the Robotino mobile robot. Algorithms written for Robotino can be tested through different sensors and actuators on the robot in the Robotino[®] SIM environment [33]. An RGB camera, infrared distance sensing sensors, and 3 encoders connected to the wheels on the Robotino can be used in the simulation environment. Different-colored obstacles can be placed at the desired location, and these obstacles can be detected with infrared sensors or an RGB camera. In order to determine the route we want the robot to follow, a line can be drawn in the desired color; the color of the line can be taught to the robot, and the line can be followed. The simulation environment is displayed in 3D on the computer screen as shown in Figure 3.

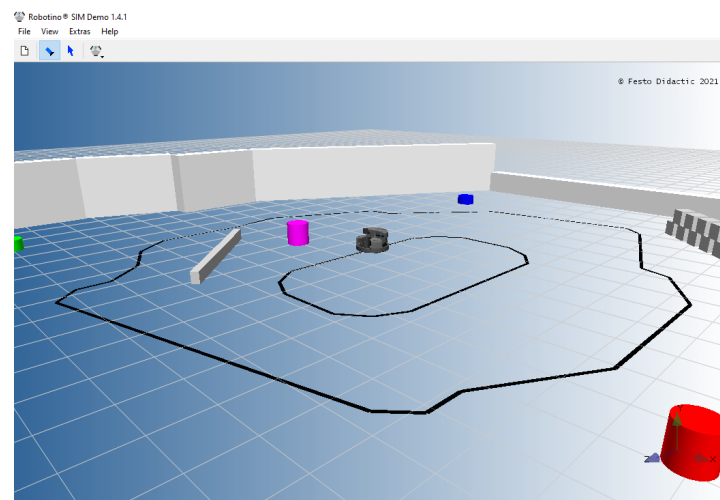


Figure 3. Three-dimensional view of RobotinoSIM.

3.5. QR Code

A QR code is a type of matrix barcode or two-dimensional code designed to store data information [34]. QR code means “quick response” in English. It was invented by the Japanese company Denso Wave. Denso Wave, a subsidiary of Toyota, used this technology for inventory tracking in the production of vehicle parts in 1994. Information is encoded in both vertical and horizontal directions, so it can store several hundred times more data than conventionally used barcodes. Data are accessed by taking an image of the code via a camera and processing the image with a QR reader [35]. In Figure 4, the stages of generating, reading, and evaluating a QR code are explained. A web page or computer executable application can be used to generate the desired QR code. In this study, the image obtained by reading the generated QR code with the camera was processed, and its meaning was determined by transferring it to the application where the code was created.

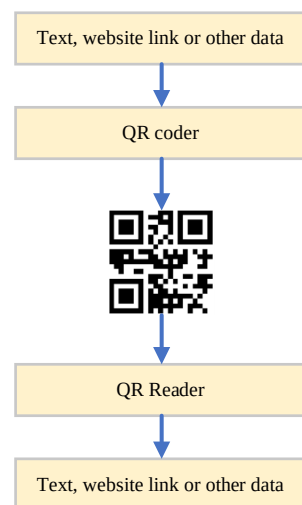


Figure 4. QR code system workflow chart.

3.6. Mapping Algorithm

The mapping algorithm enables the determination of the environment and the points where the mobile robot will work. A line tracking algorithm was written with image processing to determine the boundaries of the map. The odometry feature was used to record the path followed by the robot. To determine the working points of the robot, the QR code and the RGB camera on the Robotino were used. These operating points were then required for the mobile robot to know from where to move materials and products. The mapping algorithm works as follows. The mobile robot first records the coordinates of the starting point as x and y so that it knows from which point it started the mapping. To complete its mapping, the robot must follow the borders of the closed map and learn that it has completed the mapping when it reaches the starting point.

After recording the starting points, the robot starts to follow the map drawn as a black stripe with the line tracking algorithm. The algorithm written to read the QR code is executed while continuing to follow the line. If a QR code is detected with the camera on this path, the coordinates of the point where this QR code is detected and the meaning of the QR code are recorded. The robot continues moving after saying the name of the recording and recording point with the text-to-speech command. When the mobile robot arrives at the starting point, it completes the map of the closed environment. The mapping process is completed with a code saying, “mapping has been terminated”, to notify the user.

3.7. Path Tracing Algorithm

The path tracing algorithm works as follows. By means of odometry, the position of the mobile robot is measured pointwise on the xy plane. It is assumed that there is a straight road between the position of the given target point and the position of the current point, and this path will be followed. The x -axis of the Robotino mobile robot must coincide with the line on the current position and target point. For this, the angle in the Cartesian plane must be found between two points. This angle is calculated using Equation (10).

The instantaneous angle of the mobile robot is determined through the data obtained from the odometer. The mobile robot turns until the difference between its own angle and the target angle reaches approximately zero. When the angle is determined, the robot is now headed towards the target point from its current position. By making the robot move at a constant speed at this angle, it is ensured that it moves towards the target point. When the x and y values read from the odometry data of the mobile robot approaching the target point approximate the x and y values of the target point, the mobile robot reaches the target point and stops, as shown in Figure 5.

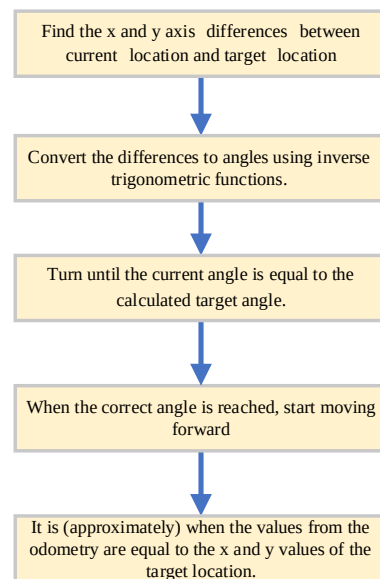


Figure 5. Flowchart of the path tracing algorithm.

3.8. Work Order Receiving and Execution Algorithm

In this study, we decided to conduct the tasks of the mobile robot with the work order method. With the work order, the mobile robot is provided to carry materials between the working areas that it has learned through mapping. A starting point and destination point are determined in the work order. In addition, the materials to be transported between the starting point and the destination point are defined. These materials to be transported are defined separately so that the mobile robot can understand whether it has received the material it is carrying through QR codes in which the materials are defined. The work order is transmitted to the mobile robot via an Excel spreadsheet. A screenshot of the Excel table where the work order is to be executed is given in Figure 6. The Excel file is backed up in the robot's memory or on the computer where the robot is controlled. Each work order line can be increased to 16 stations; unnamed stations are not in use. Each line contains the date and time information on which the work order was given and completed.

	A	B	C	D	E	F	G	H	I	J	K
1	Start Date	Start Time	Finish Date	Finish Time	Station 1	Station 2	Station 3	Station 4	Station 5	Station 6	Station 7
2	2023-02-01	10:23:05	2023-02-01	10:28:15	Loading	B	Storage	D			
3	2023-02-01	10:33:01	2023-02-01	10:36:02	A	Loading	C	Storage			
4	2023-02-01	11:03:05	2023-02-01	11:08:50	A	Storage	C	Loading			
5	2023-02-01	11:20:45	2023-02-01	11:23:15	Storage	B	C	Loading			
6	2023-02-01	11:25:11	2023-02-01	11:29:32	A	Loading	Storage	D			
7											
8											
9											
10											
11											

Figure 6. Screenshot of Excel table that allows the work order to be executed.

When a work order is started, the name of the starting point is first checked. The name of the starting point is chosen from the locations learned in the mapping, and the coordinates of this location are determined. After the points are determined, the location to be reached by the text-to-speech synthesizer is announced audibly by the mobile robot to its surroundings. Afterwards, the path tracking algorithm described in the previous sections is activated, and the mobile robot moves from its current position to its starting position. When it realizes that it has come to the location, the materials to be transported,

which are entered in the work order, are requested from the user in order. It is confirmed that the materials have been given to the robot by reading the QR code representing each desired material.

After knowing that the parts to be transported have been received, the robot proceeds towards the target point, again by subtracting the x and y points of the target point, through the path tracking algorithm. When it reaches the target point, the robot again asks the users to pick up the parts audibly. The materials are to be transported by scanning the QR code, confirmation of the receipt of the parts is obtained, and a work order is completed. These processes continue until the work order numbers are completed. After the work orders are completely completed, the mobile robot potentially goes to the next point with the machine learning algorithm and goes into the work order waiting state. When a new work order is entered, the work order ready button is pressed, and the mobile robot is again enabled to fulfill work orders.

3.9. Machine Learning Algorithm

Machine learning is an evolving computational algorithm designed to mimic human intelligence by learning from the surrounding environment [36]. A machine learning algorithm is also defined as a computational process that uses input data to perform a desired task without being programmed to produce a specific result [37].

Unsupervised learning was used in the machine learning algorithm developed within the scope of this study. An estimation algorithm was written based on the data entered with the work order via machine learning. Due to this algorithm, after completing the work order, the robot will wait at the most selected point among the points where the mobile robot takes a load. This point is determined as follows. If the robot receives the need to carry a load continuously from a specific loading point, it learns that it has received many work orders from this point, guesses that the next waiting point should be there, and waits for the new work order at this point. However, if the work order comes from another point rather than the loading point, it predicts that there is a density at that point and determines the waiting point there.

For this algorithm to work, all work orders must be recorded. Then, the most repetitive work order is determined among these recorded data. The flow chart of the main algorithm, which is formed by combining these algorithms, is shown in Figure 7.

To use the mobile robots for material transport between production stations in smart factories, first, a black line was drawn between the production stations, a QR code was attached to each station, and the station was named. When the mobile robot is placed at the determined starting point, it firstly follows the black line with the RGB camera on the robot, reads the QR codes on all production stations, and saves the coordinates of these stations with their names. When it reaches the starting point after all QR codes are read, the mapping and placement process is complete. The robot tells the user audibly when it has started the mapping process. It notifies the user audibly when the mapping or placement process is completed at each production station it determines. Since the mapping and placement process is made by a supervisor to teach the robot about the working environment, it does not include obstacle avoidance procedures, as there is no possibility of an obstacle or unknown object in the working environment.

When the mobile robot receives the work order, it calculates the movement angle using the coordinate information of the station where it is located and the target station, the station where it will receive the material, and starts moving according to the angle. The robot reaches the target station using the shortest path between these two stations. When it encounters an obstacle that is not included in the mapping and placement process in the work area, it reaches the target by wandering from the right or left of the obstacle without extending its path. It uses the RGB camera image to detect the obstacle. If the color of the obstacle is the same as the color of the working area, then the infrared sensors on the robot allow it to detect the obstacle.

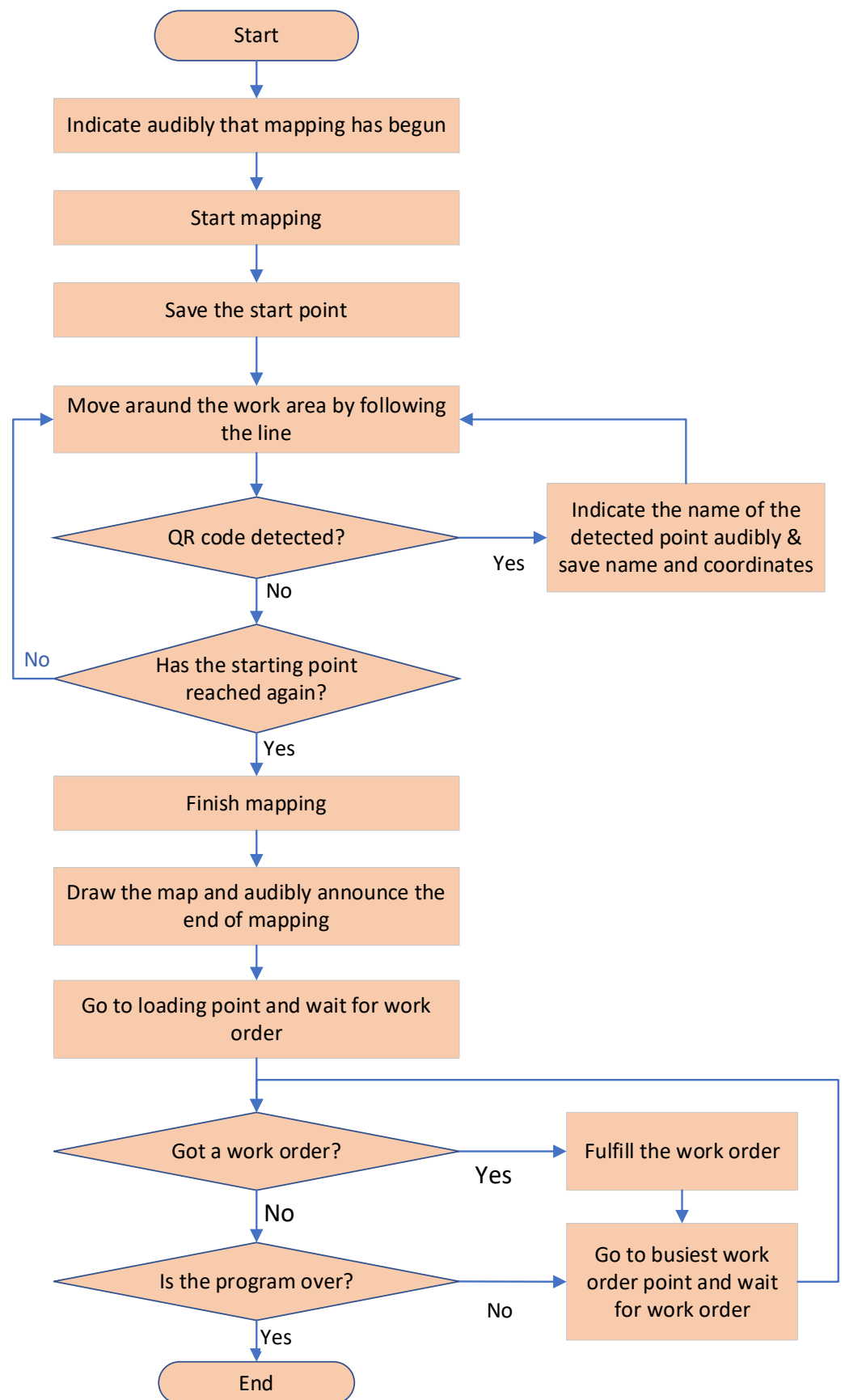


Figure 7. Flowchart of the main algorithm.

When it reaches the destination station, it receives the load and similarly goes to the destination station in the shortest way and completes the work order. When the robot reaches the target station, it reads the QR code and determines the coordinates of the target. If there is no work order waiting in the queue, it waits at a station until a new work order arrives. While determining the waiting station, it waits at the station where it received the most work orders before. If there are stations with an equal number of work orders, it waits at the nearest station to minimize energy consumption.

4. Experimental Study

The test of the developed algorithms was firstly performed in the Robotino[®] SIM demo program to be safe and respond quickly. Then, we uploaded and tested the entire algorithm to work on the Festo Robotino in the Robotics Laboratory in Ege University since it is not possible to read the QR code and the physical angle of the robot using the simulation program. The sub-algorithms required for the whole algorithm were designed and tested with Robotino[®] SIM Demo. These algorithms are the mapping algorithm mentioned in Section 2, the text-to-speech synthesizer algorithm, the path tracing algorithm to go from one point to another, and the algorithm required to receive the work order and implement it.

Since the QR code usage cannot be tested in the study with Robotino[®] SIM, the locations and names of the working points were entered manually. After mapping, Robotino was run between these locations according to a given scenario. The robot first creates the map and then goes to the specified production station. In the given scenario, four production stations were determined and entered, as shown in Figure 8a. The robot first went to the loading point, as shown in Figure 8b, and then it went to station 1, as shown in Figure 8c. In accordance with the given scenario, it then went to station 2, as shown in Figure 8d, and finally went to the storage station, as shown in Figure 8e, and completed its task there.

For the physical test in the robotics laboratory, a working environment for the mobile robot was prepared with black tape, as shown Figure 9. To test the workability of this working environment line tracing algorithm, an indented map was chosen. QR codes were placed in the environment, whose map was drawn with a black band, to determine the locations of the working points, as shown Figure 10a. The names may vary, and three points were determined as the operating points for the test. These points were for the loading algorithm; they were named as the number of working points, the locations of the working points, the point of the working points, station 1, and station 2. The materials used for transport were representatively determined by QR codes. As shown in Figure 10b, the robot took a picture of the QR code and compared this code with the given code in the program. These QR codes are shown with alphabetic letters, such as A, B, C, and D, and the products formed after the process are represented by a combination of letters such as ABC, BD, and AC.

During the physical test, mapping, the determination of working points, and the operation of the mobile robot between these working points in accordance with the work order were observed.

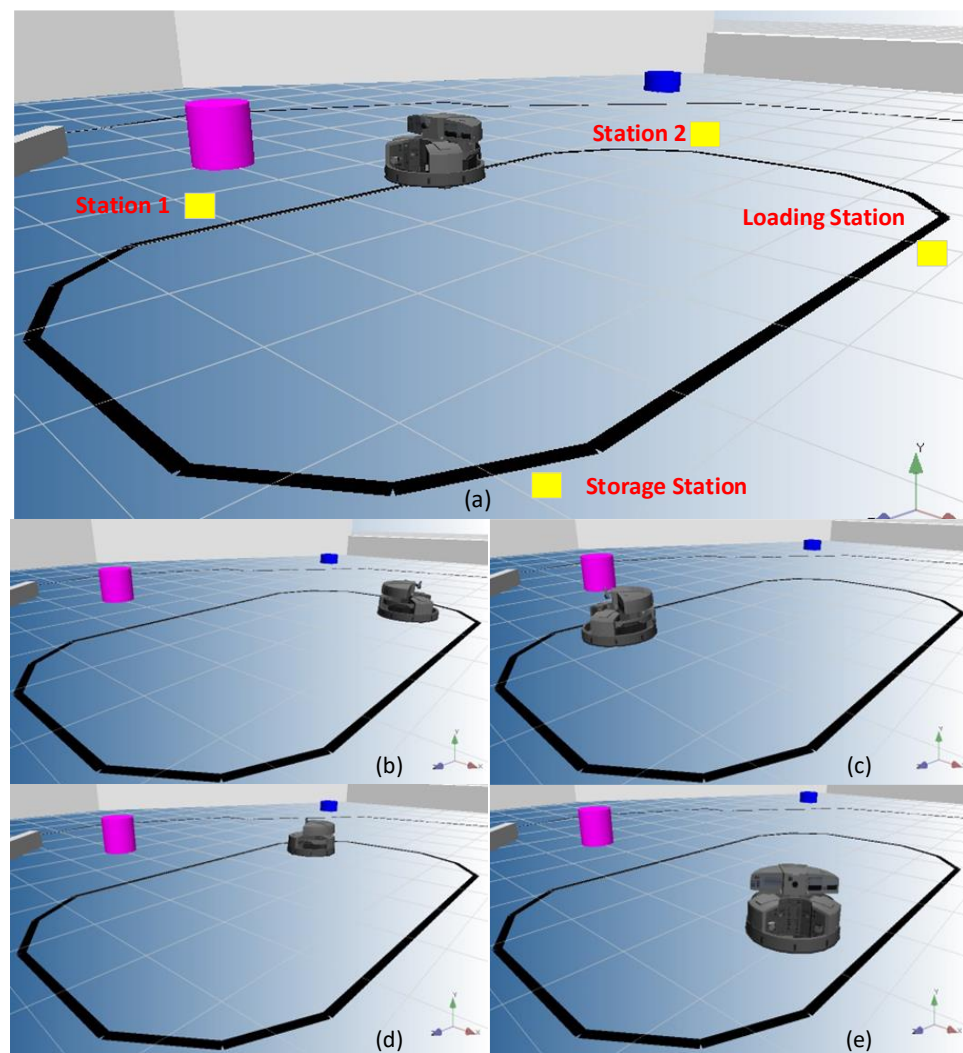


Figure 8. (a) Locations determined in the simulation. (b) The mobile robot is located at the loading station. (c) The mobile robot is located at station 1. (d) The mobile robot is located at station 2. (e) The mobile robot is located at the storage station.



Figure 9. Workspace prepared for physical testing.

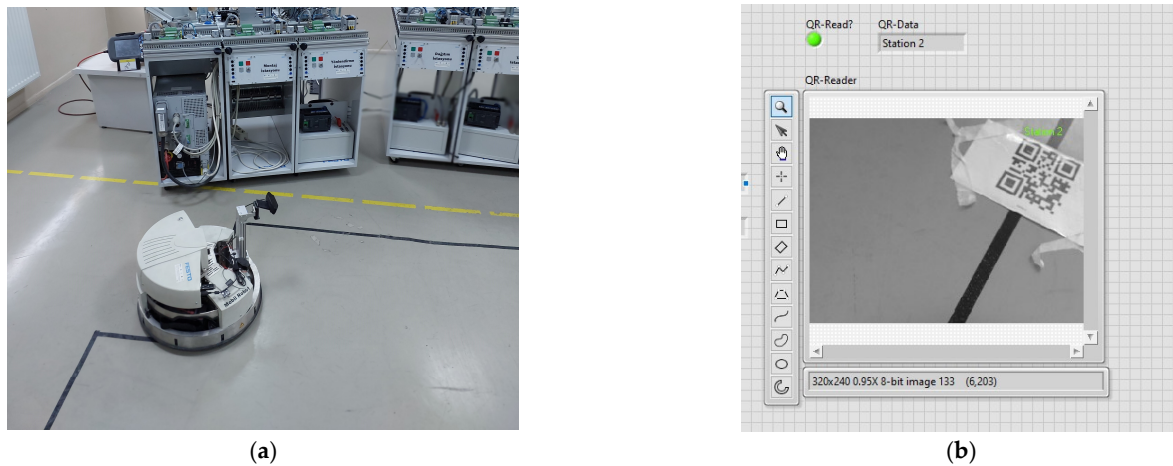


Figure 10. (a) In the case of mobile robot mapping. (b) QR code reader screenshot during physical test.

5. Results and Discussion

5.1. Results

In the simulation phase, the mapping algorithm, the text-to-speech synthesizer algorithm, the operation between manually entered locations, and the work order receiving and fulfillment algorithm were tested. As a result of the simulation, the mobile robot mapped the working area through odometry and the line tracking algorithm. In the previous sections of the article, the visuals of the area to be mapped were given. As a result of the mapping, the map of the work area is given in Figure 11a. The accuracy of this map could not be measured due to the lack of the capacity of the simulation program in this regard, but when the map and the work area were visually checked, the coordinates of the map were determined correctly. Another test made in the simulation program was the correct operation of the work order receiving and fulfillment algorithm and the text-to-speech synthesizer. The mobile robot fulfilled the work orders taken from the Excel file, respectively, and there was no problem with the text-to-speech algorithm; the algorithm worked correctly.

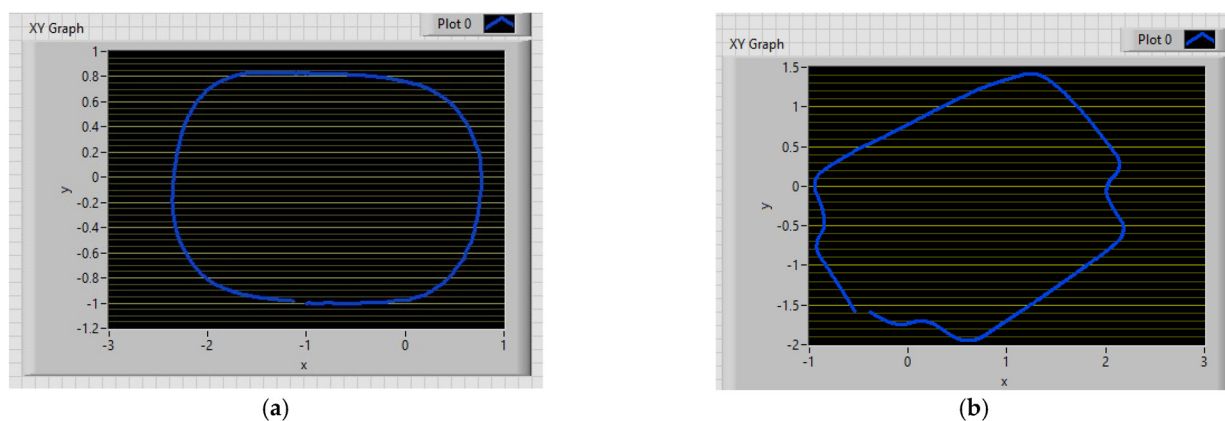


Figure 11. (a) Map of the workspace in the Robotino[®] SIM program; (b) map of the physical test workspace.

In the study carried out in the robotics laboratory, the results obtained in the simulation were again obtained. During the physical test, the mobile robot was able to map the given work area by following the black line. It was able to record the working points by reading the QR codes on the road. As a result of a few tests, the QR codes used in the dimensions that the camera could see were easily viewed through the camera of the mobile robot, and the code was deciphered. However, depending on the light intensity of the test

environment, when the light intensity decreased, there was a negative effect on the reading of the QR code, and in some trials, the robot could not fully detect the QR code.

When the mobile robot completes the mapping and reaches the starting point, it has finished the mapping. When the map was drawn, sharp corners could not be observed on the map drawn due to the line tracing algorithm, but the boundaries of the work areas were clearly determined, as shown in Figure 11b.

After the mapping was completed, the mobile robot was able to fulfill the work orders. It successfully read the QR codes used to introduce the materials to be transported and was able to detect the materials given to and taken from the robot. After mapping, the robot saved the names of the production stations, the coordinates on the x -axis, and the coordinates on the y -axis as separate files. These files are shown in Figure 12a, b, and c, respectively, and can be accessed whenever needed.

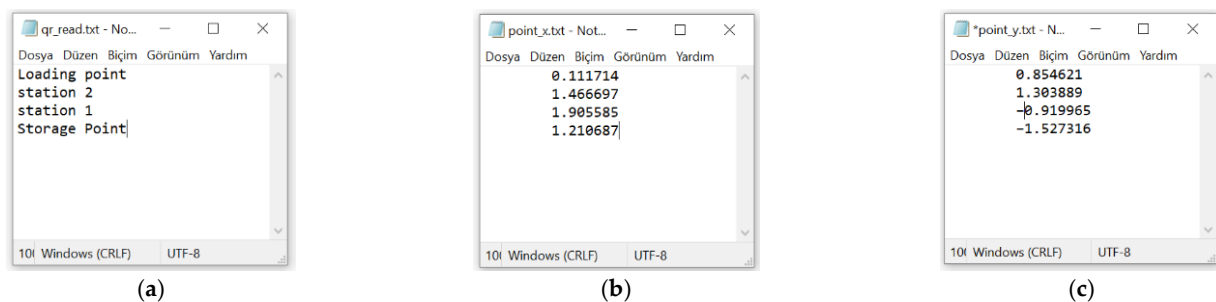


Figure 12. (a) Sequential names of operating points; (b) coordinates on the x -axis of the working points; (c) coordinates on the y -axis of the working points recorded during the physical test.

The mapping algorithm developed in this article can be quickly rerun, and a new work map can be obtained when the task to be given to the robot in the smart factory facility is changed or a new production station is added or removed. The fact that the map is easily redefinable makes it possible to use this type of robot in flexible production systems, which are widely preferred in today's factories.

To test the machine learning algorithm, we created 50 random work orders consisting of 6 stations. We recorded the total distance and total time the robot traveled while executing these work orders. The operating system of the robot records these values in normal operation, and the user can read these values at any time.

A total of 50 work orders were programmed according to 3 different waiting station usage scenarios, and the results obtained by running them are given in Table 1. In the first scenario, the robot waited at the station determined as the home position after completing the first order. In this experiment, station 1 was set as the home position. The robot started the second work order from the home position. In the second scenario, the robot waited for the next work order at the last station after completing the work order. In the third scenario the robot waited at the station where the most work orders were given according to the machine learning algorithm.

Table 1. Experimental result of three different waiting station usage scenarios.

Waiting Station	Total Duration	Total Distance Traveled	Success
Home Position	190 min 56 s	585 m	100%
Finish Position	180 min 24 s	558 m	100%
Machine Learning Algorithm	150 min 20 s	450 m	100%

When the task is given after mapping, it performs the task by calculating the shortest path to use the least energy on the way to the production station. Low energy consumption is important for two reasons. The first one is that the robot can work for a longer time without the need for charging since it works with a battery and has a high working efficiency.

Secondly, the small carbon footprint during operation is a feature expected from every production facility today.

5.2. Discussion

The use of the QR code allowed the names of the production stations to be registered correctly and to be modified as needed. In addition, the developed QR code reading program can also be used for naming the loads to be transported, thus preventing the wrong load from being transported.

During this study, the working area of the robot was free from unknown objects, and therefore, the obstacle avoidance algorithm was not prepared. However, by using the image processing algorithm using an RGB camera, obstacles in the working area were detected, and the robot was able to rearrange its motion route in real time. It is sufficient that the color of the object in the area is different from the color of the ground. As shown in Figure 8a–g, the algorithm developed by placing obstacles of assorted colors in the work area in the simulation environment has been tested. In addition, owing to the infrared sensors on the robot, if the robot comes closer to unknown objects more than 3 cm, it stops the robot's movement and prevents a collision.

Vision-based SLAM is widely used for navigation of agricultural, industrial automation, medical, logistics, aerospace, and military robots. In the literature, researchers have carried out some of the mapping, localization, collision avoidance, obstacle detection, and path planning stages of navigation as vision based. In dynamic environments especially, vision-based applications for mapping and collision prevention offer a cheaper and successful solution compared to other methods. When there is a problem with our eyes in our daily life, it negatively affects our movements. This is also true for the service robots that will work with humans; the vision-based method is as effective as our eyes. While most researchers have solved localization and mapping in robot navigation with vision-based sensors, they have proposed other sensors for the obstacle detection and path planning stages [38]. A proposal that offers solutions to all stages of robot navigation with only a vision-based sensor has not been studied in the literature. Our work validates the entire mapping, placement and collision avoidance stages, and the path planning stage with a vision-based sensor. In addition to navigation, it also uses the same sensor in object handling. When evaluated from these perspectives, it will be the first in the literature.

When the appropriate environmental conditions were provided, the robot performed the given mapping, placement, collision avoidance, and product transportation processes with full success. At the beginning of the constraints that prevent success, the main problem in all vision-based methods is that the ambient light level is insufficient or too much. According to the results of our experimental studies, if the light level in the environment is less than 100 lumens, the robot's success of mapping decreases to 50%, while QR code reading is reduced to 25% and collision prevention to 65%. No adverse effects on mapping and collision avoidance have been observed at a light level above 1200 lumens. However, it has been observed that it reduces the level of success in reading QR codes, especially in reading QR codes printed on white paper, to 70%. Bai et al. found that in vision-based SLAM studies in agricultural applications, low light levels reduce success without giving any numerical value [21]. They determined that high-level light, especially horizontally incoming sun rays, causes shading and adversely affects success.

With the vision-based sensor we used, the fact that, in the collision avoidance process, the color of the unknown object entering the working environment is very close to the ambient color or other production tools in the environment reduces the level of success. Similarly, the fact that the color of the object to be transported is the same as the environment in which it is located adversely affects the success. To solve this problem, our recommendation is to use the binocular vision method where we can obtain both visual data and depth data. However, although this method has a high performance level, it has some application limitations, such as real-time operation with latency acceptable only for some applications and high computational complexity. The RGB-D is a newly developed

visual sensor that has been introduced in recent years. When exposed to high levels of sunlight in open environments, the success level of this sensor decreases [39]. Since our application is not likely to be exposed to direct sunlight in the indoor environment, this low-cost sensor will be a solution with a high level of success.

If it encounters an obstacle during the transport operation, it calculates the shortest path around the obstacle and completes its task. If there is no new work order in the queue, it waits until a new work order arrives at the station where it receives the most work orders. The results of our experimental study showed that waiting at the station where the robot received the most work orders resulted in 20% less travel time than waiting at the nearest station and 27% less travel time than waiting at a fixed station. The energy consumed by the robot will decrease in direct proportion to the distance it travels. The robot completed the given 50 work orders with 100% accuracy.

During the mapping and location process of the robot, odometry calculation errors caused by slips on the wheels, and errors during processing the image from the RGB camera are added, and these errors can increase to unacceptable levels in the work order. Each time the robot reaches the station the QR code is read, and these errors are reset before they increase.

6. Conclusions

In this study, a method is proposed for the use of a mobile robot in a smart factory. A mobile robot was used for operations such as transporting raw materials to the production line, recycling any product that did not meet standards, and transporting the final product to the warehouse.

First, the robot performed simultaneous mapping and localization of the work area under the supervision of a supervisor using an RGB camera and a QR code attached to each station. After the mapping and localization process, it waited for the first work order at the starting point. After receiving the work order, it reached the target station using the shortest path it could take from its current location. The algorithm verified the coordinate of the target by reading the QR code at the station. After receiving the product, the robot reached the station where it would leave the product by the shortest route and completed the work order.

Although our method has good performance, it also has some limitations. Two types of problems arise when a vision-based sensor is used in robot navigation. The first is that at very low or very high ambient light levels, the sensor fails to collect enough data. The second is that the color of unknown objects in the collision avoidance study in dynamic environments is not perceived correctly because it is the same as the color of the environment or is very close.

Our possible future research topic may be to use an RGB-D sensor in mobile robot navigation. Thus, we can eliminate the detection errors of unknown objects in dynamic environments that are the same color as the environment, which is the problem we encountered in our research. Another future research goal is to navigate two or more mobile robots simultaneously in dynamic environments in smart factory applications.

Author Contributions: Conceptualization, M.E. and N.K.; methodology, M.E.; software, N.K.; validation, M.E. and N.K.; formal analysis, N.K.; investigation, N.K.; resources, M.E.; data curation, N.K.; writing—original draft preparation, N.K.; writing—review and editing, M.E.; visualization, N.K.; supervision, M.E.; project administration, M.E.; funding acquisition, M.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Rubio, F.; Valero, F.; Llopis-Albert, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419839596. [\[CrossRef\]](#)
- Khan, M.G.; Huda, N.U.; Zaman, U.K.U. Smart Warehouse Management System: Architecture, Real-Time Implementation and Prototype Design. *Machines* **2022**, *10*, 150. [\[CrossRef\]](#)
- Tong, Q.; Ming, X.; Zhang, X. Construction of Sustainable Digital Factory for Automated Warehouse Based on Integration of ERP and WMS. *Sustainability* **2023**, *15*, 1022. [\[CrossRef\]](#)
- Fatima, Z.; Tanveer, M.H.; Waseemullah; Zardari, S.; Naz, L.F.; Khadim, H.; Ahmed, N.; Tahir, M. Production Plant and Warehouse Automation with IoT and Industry 5.0. *Appl. Sci.* **2022**, *12*, 2053. [\[CrossRef\]](#)
- Gonzalez-Aguirre, J.A.; Osorio-Oliveros, R.; Rodríguez-Hernández, K.L.; Lizárraga-Iturralde, J.; Menendez, R.M.; Ramírez-Mendoza, R.A.; Ramírez-Moreno, M.A.; Lozoya-Santos, J.D.J. Service Robots: Trends and Technology. *Appl. Sci.* **2021**, *11*, 10702. [\[CrossRef\]](#)
- Holland, J.; Kingston, L.; McCarthy, C.; Armstrong, E.; O'Dwyer, P.; Merz, F.; McConnell, M. Service Robots in the Healthcare Sector. *Robotics* **2021**, *10*, 47. [\[CrossRef\]](#)
- Wang, S.; Wang, K.; Tang, R.; Qiao, J.; Liu, H.; Hou, Z.-G. Design of a Low-Cost Miniature Robot to Assist the COVID-19 Nasopharyngeal Swab Sampling. *IEEE Trans. Med. Robot. Bionics* **2021**, *3*, 289–293. [\[CrossRef\]](#) [\[PubMed\]](#)
- Lee, H.; Jeong, J. Mobile Robot Path Optimization Technique Based on Reinforcement Learning Algorithm in Warehouse Environment. *Appl. Sci.* **2021**, *11*, 1209. [\[CrossRef\]](#)
- Belanche, D.; Casaló, L.V.; Flavián, C.; Schepers, J. Service robot implementation: A theoretical framework and research agenda. *Serv. Ind. J.* **2020**, *40*, 203–225. [\[CrossRef\]](#)
- Tofangchi, S.; Hanelt, A.; Marz, D.; Kolbe, L.M. Handling the Efficiency–Personalization Trade-Off in Service Robotics: A Machine-Learning Approach. *J. Manag. Inf. Syst.* **2021**, *38*, 246–276. [\[CrossRef\]](#)
- Groover, M.P. *Automation, Production Systems, and Computer-Integrated Manufacturing*, 5th ed.; Pearson: Chennai, India, 2016.
- Nivas, V.M.; Krishnan, P.G.; Fredrhc, A.C. Automated Guided Car (AGC) for industrial automation. In Proceedings of the 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, India, 24–26 February 2016; pp. 1–6. [\[CrossRef\]](#)
- Hussain, R.A.A.; Mohd-Mokhtar, R. Development of a Logistics Autonomous Mobile Robot (EasyBot). In *Lecture Notes in Electrical Engineering*; Springer: Singapore, 2022; Volume 770, pp. 1009–1025. [\[CrossRef\]](#)
- Mulyana, T.; Rachmat, H.; Yuliarso, P.P. An Automated Guided Vehicle Simulation through Robotino to Help Learning Course Industrial Automation. In Proceedings of the 9th International Seminar on Industrial Engineering and Management, Padang, Indonesia, 20–22 September 2016; pp. 38–44. Available online: https://isiem.net/wp-content/uploads/2016/10/9th_ISIEM_2016_paper_33_ps_Proceeding.pdf (accessed on 11 March 2023).
- Di Paola, D.; Milella, A.; Ciciirelli, G.; Distanti, G.C.A.A. An Autonomous Mobile Robotic System for Surveillance of Indoor Environments. *Int. J. Adv. Robot. Syst.* **2010**, *7*, 8. [\[CrossRef\]](#)
- Anil, A. Ulusal Tez Merkezi | Anasayfa. Design and Control of a Mobile Autonomous Library Robot. 2019. Available online: https://tez.yok.gov.tr/UlusalTezMerkezi/tezDetay.jsp?id=RKNLg_j824Tn9ypepbNPYA&no=tXxm3pkChXRv2bAjYtyosw (accessed on 11 March 2023).
- Rothomphiwat, K.; Harnkhamen, A.; Tothong, T.; Suthisomboon, T.; Dilokthanakul, N.; Manoonpong, P. Advanced Collaborative Robots for the Factory of the Future. In Proceedings of the 2021 IEEE/SICE International Symposium on System Integration (SII), Iwaki, Japan, 11–14 January 2021; pp. 578–579. [\[CrossRef\]](#)
- Chan, T.H.; Hesse, H.; Ho, S.G. LiDAR-Based 3D SLAM for Indoor Mapping. In Proceedings of the 2021 7th International Conference on Control, Automation and Robotics (ICCAR), Singapore, 23–26 April 2021; pp. 285–289. [\[CrossRef\]](#)
- Hanagi, R.R.; Gurav, O.S.; Khandekar, S.A. SLAM using AD* Algorithm with Absolute Odometry. In Proceedings of the 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2–4 April 2021; pp. 1–4. [\[CrossRef\]](#)
- Tsintotas, K.A.; Sevetlidis, V.; Papapetros, I.T.; Balaska, V.; Psomoulis, A.; Gasteratos, A. BK tree indexing for active vision-based loop-closure detection in autonomous navigation. In Proceedings of the 2022 30th Mediterranean Conference on Control and Automation (MED), Vouliagmeni, Greece, 28 June–1 July 2022; pp. 532–537. [\[CrossRef\]](#)
- Bai, Y.; Zhang, B.; Xu, N.; Zhou, J.; Shi, J.; Diao, Z. Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review. *Comput. Electron. Agric.* **2023**, *205*, 107584. [\[CrossRef\]](#)
- Narazaki, Y.; Hoskere, V.; Chowdhary, G.; Spencer, B.F. Vision-based navigation planning for autonomous post-earthquake inspection of reinforced concrete railway viaducts using unmanned aerial vehicles. *Autom. Constr.* **2022**, *137*, 104214. [\[CrossRef\]](#)
- Adamkiewicz, M.; Chen, T.; Caccavale, A.; Gardner, R.; Culbertson, P.; Bohg, J.; Schwager, M. Vision-Only Robot Navigation in a Neural Radiance World. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4606–4613. [\[CrossRef\]](#)
- Didactic, F. Mobile Robotics—Focus & Trending Topics I4.0—Factory Automation & Industry 4.0—Learning Systems—Festo Didactic. 2023. Available online: <https://www.festo-didactic.com/int-en/learning-systems/factory-automation-industry-4.0/focus-trending-topics-i4.0/858/?fbid=aW50LmVuLjU1Ny4xNy4yMC44NTg> (accessed on 11 March 2023).
- Didactic, F. Robotino® View2 EN. 2023. Available online: www.festo-didactic.com (accessed on 11 March 2023).
- Matlab—RobotinoWiki. 2023. Available online: http://wiki.openrobotino.org/index.php?title=Matlab#Running_Robotino_MATLAB_drivers_with_32_bit_MATLAB (accessed on 11 March 2023).

27. Labview—RobotinoWiki. 20 September 2023. Available online: http://wiki.openrobotino.org/index.php?title=Labview#LabVIEW_Drivers_available (accessed on 11 March 2023).
28. Bayar, G.; Ozturk, S. Investigation of The Effects of Contact Forces Acting on Rollers Of a Mecanum Wheeled Robot. *Mechatronics* **2020**, *72*, 102467. [CrossRef]
29. Liu, Y.; Zhu, J.J.; Williams, R.L.; Wu, J. Omni-directional mobile robot controller based on trajectory linearization. *Robot. Auton. Syst.* **2008**, *56*, 461–479. [CrossRef]
30. Kodosky, J. LabVIEW. *Proc. ACM Program. Lang.* **2020**, *4*, 1–54. [CrossRef]
31. Azis, N.A.; Hikmah, R.; Mjahja, T.; Nugroho, A.S. Evaluation of text-to-speech synthesizer for Indonesian language using semantically unpredictable sentences test: IndoTTS, eSpeak, and google translate TTS. In Proceedings of the 2011 International Conference on Advanced Computer Science and Information Systems, Jakarta, Indonesia, 17–18 December 2011; pp. 237–242. Available online: <https://ieeexplore.ieee.org/abstract/document/6140759> (accessed on 12 March 2023).
32. Panoiu, M.; Rat, C.-L.; Panoiu, C. A comparative study of text-to-speech systems in LabVIEW. In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2016; Volume 356, pp. 3–11. [CrossRef]
33. Simulation—Robotino®—Services—Festo Didactic. Available online: <https://www.festo-didactic.com/int-en/services/robotino/simulation/?fbid=aW50LmVuLjU1Ny4xNy4zNC4xNDQy> (accessed on 12 March 2023).
34. Tiwari, S. An Introduction to QR Code Technology. In Proceedings of the 2016 International Conference on Information Technology (ICIT), Bhubaneswar, India, 22–24 December 2016; pp. 39–44. [CrossRef]
35. Kieseberg, P.; Leithner, M.; Mulazzani, M.; Munroe, L.; Schrittwieser, S.; Sinha, M.; Weippl, E. QR code security. In Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia, Paris, France, 8–10 November 2010; pp. 430–435.
36. Zincir-Heywood, N.; Mellia, M.; Diao, Y. Overview of Artificial Intelligence and Machine Learning. In *Communication Networks and Service Management in the Era of Artificial Intelligence and Machine Learning*; Wiley Online Library: Hoboken, NJ, USA, 2021; pp. 19–32. [CrossRef]
37. Bell, J. What Is Machine Learning? In *Machine Learning and the City*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 2022; pp. 207–216. [CrossRef]
38. Yasuda, Y.D.V.; Martins, L.E.G.; Cappabianco, F.A.M. Autonomous Visual Navigation for Mobile Robots. *ACM Comput. Surv.* **2020**, *53*, 1–34. [CrossRef]
39. Skoczeń, M.; Ochman, M.; Spyra, K.; Nikodem, M.; Krata, D.; Panek, M.; Pawłowski, A. Obstacle Detection System for Agricultural Mobile Robot Application Using RGB-D Cameras. *Sensors* **2021**, *21*, 5292. [CrossRef] [PubMed]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.