Turkish Journal of Science & Technology
17(1), 37-46, 2022

Research Paper
https://doi.org/10.55525/tjst.1054213

Mobile Robot Indoor Localization Using Color-Coded Beacons and a Depth Camera

Mawj MOHAMMED BASHEER[1*], Mehmet ÇAVAŞ[2], Mohammed QASIM[3]
[1,2] Mechatronics Engineering, Technology Faculty, Firat University, Elazığ, Turkey
[3]Systems & Control Engineering, Electronics engineering Faculty, Ninevah University, Mosul, Iraq
[*1] eng.mawj.aljadirjy@gmail.com, [2] mcavas@firat.edu.tr, [3]mohammed.qasim@uoninevah.edu.iq

(Geliş/Received: 11/01/2022; Kabul/Accepted: 11/02/2022)

Abstract: This paper aims to design and implement an indoor localization method for mobile robots based on trilateration technique, Color Code signatures (CCs) as artificial passive beacons, and a depth camera. The artificial passive beacons are designed as three disks positioned one on top of the other’s with different sizes and colors (primary colors only, RGB). The three disks share the same center. The designed beacons are placed on the ceiling to be visible from most locations within the room. The robot's model, controller, and localization method are implemented and evaluated inside the CoppeliaSim environment. The simulation results show that the CCs detecting algorithm enables the robot to discover the designed beacons, achieve an accurate localization and reach the target.

Keywords: Omnidirectional robot, localization, CoppeliaSim program, Trilateration.

Renk Kodlu İşaretler ve Derinlik Kamerası Kullanan Mobil Robot İç Mekan Lokalizasyonu

Öz: Bu makale, mobil robotlar için trilaterasyon tekniğine, yapay pasif işaretler olarak Renk Kodu imzalarına (CC'ler) ve bir derinlik kamerasına dayalı bir iç mekan lokalizasyon yöntemi tasarlamayı ve uygulamayı amaçlamaktadır. Yapay pasif işaretçiler, farklı boyut ve renklerde (yalnızca ana renkler, RGB) biri diğerinin üzerine yerleştirilmiş üç disk olarak tasarlanmıştır. Üç disk aynı merkezi paylaşır. Tasarlanan fenerler, oda içindeki çoğu yerden görülebilecek şekilde tavana yerleştirilmiştir. Robotun modeli, denetleyicisi ve yerelleştirme yöntemi CoppeliaSim ortamında uygulanır ve değerlendirilir. Simülasyon sonuçları, CC'lerin tespit algoritmasının robotun tasarlanan işaretleri keşfetmesini, doğru bir lokalizasyon elde etmesini ve hedefe ulaşmasını sağladığını göstermektedir.

Anahtar kelimeler: Omnidirectional robot, yerelleştirme, CoppeliaSim programı, Trilaterasyon.

1. Introduction

Robotics is an interdisciplinary field that consists of the integration of computer science, electronics, artificial intelligence, mechanics, and control. Mobile robots are the type of robots that can move around and are not fixed in a single physical location [1]. Many studies deal with mobile robots and how they can move to achieve their objectives in the real world without the need for humans. For the correct operation of mobile robots, several technological areas and fields should be noted and integrated to understand the fundaments such as the locomotion system and kinematics, perception system (sensors), localization system, and navigation system. All these systems must be followed by a control system in order to make the mobile robot achieves its action or task in an intelligible way [2]. Localization is considered one of the essential problems that mobile robots face. The best explanation of the localization term is "the ability of the mobile robot to recognize its location within the environment." In other words, the localization term refers to the position and orientation of a mobile robot within a specific location [3]. In order to solve the localization problem, there are different practical applications depending on passive or active beacons. Beacons make the optimal localization operation easy. Despite, the artificial beacons are necessary to achieve localization, the number of beacons that are placed in the environment is limited. For the previous reason, a localization algorithm for mobile robots is used with specific placement and number of beacons [9]. The trilateration method needs at least three beacons to be recognized at the same time, with the distance between them and the robot to work[10]. Since advanced robots are still highly expensive and can be damaged in the workspace, simulators are used to create virtual environments which have many advantages for robotics studies. The CoppeliaSim software is one of the most widely used simulators today, and it deserves special attention. This simulator is a scalable and flexible framework for creating 3D simulations in a short time. CoppeliaSim includes

[*] Corresponding author: eng.mawj.aljadirjy@gmail.com.ORCID Number of authors: [1] 0000-0003-1687-6835, [2] 0000-0002-0130-1644, [3]0000-0003-2947-8887

a large number of examples, robot models, sensors, and actuators, which can be used to build a virtual environment and interact with it in real-time. To execute customized simulation experiments, new models can be built and added to the CoppeliaSim program [8]. for these reasons, the CoppeliaSim simulation platform has been used. Artificial landmarks are often installed at a predefined area of the robot's workspace. the robot needs sensors to detect or measure the angle and distance to these landmarks. Measuring distance is simple for many sensors, such as cameras or wireless devices [11].

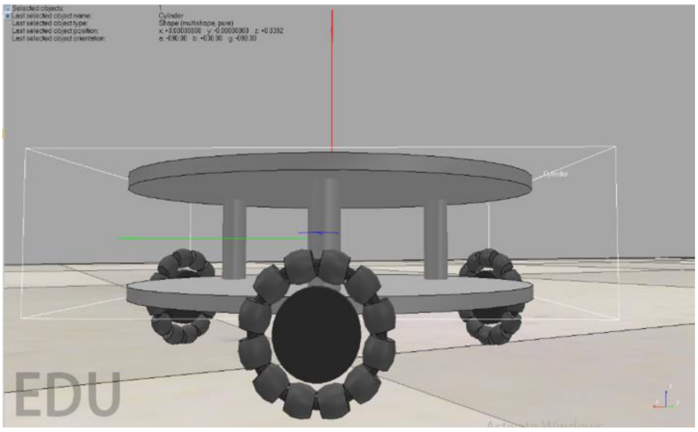
Over the past years, a wide range of research has been explored, evaluated, and applied to mobile robots to overcome the problem of localization. Chunag, L. [4] utilized RFID (Radio-Frequency Identification) tags as artificial landmarks to perform a mobile robot's localization approach. The RFID tags have been set in a well-known area within the robot's workspace, and the robot's dual-antenna RFID reader has successfully communicated with them. A similar technique was used by Chunag, L. [4] and Gueaieb et al. [5], although they used a different strategy, placing the RFID tags on the ceiling. The main problem with this technology is that exterior items (metal objects or even humans) affect indoor RF propagation; it is extremely sensitive to these items, causing measurements to be changed. Various RFID tags also result in different behaviors. As a result, if this issue is not fixed, the pre-calibrated system may be rendered unusable. As a result, infrared technology has been used as beacons to lead the blind robot through an unfamiliar environment [5]. Huosheng Hu and Dongbing Gu [12] made a study on artificial landmarks and laser scanners for localization. With six identically sized landmarks in the environment, they utilized a 360° scanner laser with a range of 50m and a constant velocity of 2Hz. For successful detection at large distances, the landmarks were made from a single strip and pre-stored in the memory. These artificial landmarks can provide powerful signals back to the scanner. When the mobile robot moves on a non-smooth floor surface. It causes the laser scanner to vibrate, resulting in incorrect readings. The infrared beam-based mobile robot localization system was first presented in 1992. Mobile robot localization applications based on infrared technology are also widely utilized for mobile localization. Before looking at how to use this technology to locate a robot, it's necessary to understand that it's divided into two categories: active beacons and passive beacons. Active beacons are those that produce infrared signals, whilst passive beacons are those that act as reflectors [13]. Recently, in 2021 [6], the researchers used Landmarks Exploration Algorithm (LEA) for indoor localization, which operates in two stages. In the first stage, the artificial color-coded landmarks (passive cylindrical landmarks) are detected and their locations are stored. At the same time, an extended Kalman filter (EKF) is used to update the robot's state. In the second stage, a proximity sensor is utilized to calculate the distance to the detected CCLs and apply the trilateration method to localize the robot. In addition, they used the CoppeliaSim environment connected with Matlab for the simulation.

The contributions of this paper include: i) unlike other methods, the approach in this paper uses passive beacons of circular shapes, which do not require a power source and are easy to build and scale (more beacons can be generated by increasing the number of rings forming them), ii) one sensor (depth camera) is all that is required to accurately localize the robot in an indoor environment and iii) the effectiveness of the proposed localization method is verified using the realistic CoppeliaSim robot simulator.

This paper is organized as follows: Section 2 illustrates the robot design in CoppeliaSim. Section 3 derives the trilateration formula. Section 4 demonstrates the design and detection of the color-coded beacon. Section 5 shows the simulation results and briefly discusses them. Section 6 concludes the papers.

2. Kinematics Modelling of Three-Wheels Omnidirectional Robot in CoppeliaSim

The Omni-Wheels Robot building is done by adding two-cylinder shapes with a diameter of 0.14 m and thickness of 0.005 m as the base of the Omni-Wheels Robot. Three Omni-wheels with a diameter of 0.0352 m and 120 degrees between each wheel joint are attached to the base. The distance between the center of the robot and the center of each wheel is $l = 0.093$ m. The final model of the Omni-Wheels robot is shown in Figure 1.



Selected objects: 1
Last selected object name: Cylinder
Last selected object type: Shape (multishape, pure)
Last selected object position: x: +0.00000000 y: -0.00000000 z: +0.0392
Last selected object orientation: a: -090.00 b: +030.00 g: -090.00

Figure 1. Omnidirectional Robot Model.

The desired angular velocities (ω_1, ω_2, ω_3) of the robot wheels can be obtained by using the following equations:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & l \\ 0 & 1 & l \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & l \end{bmatrix} \begin{bmatrix} \dot{X}_R \\ \dot{Y}_R \\ \dot{\theta} \end{bmatrix} \tag{1}$$

where r represents the radius of the wheels $\dot{X}_R$, $\dot{Y}_R$ and $\dot{\theta}$ represent the robot linear velocities along the robot X_R, Y_R axes and angular velocity around its Z_R axis respectively. The wheels' motors are then programmed to track the appropriate angular velocities using a low-level PI controller. The velocity equations are programmed by using Lua in CoppeliaSim's script.

3. Trilateration Technique

Basically, localization means that the robot has to determine its position and orientation in the environment[14]. The position will be estimated based on the concept of the trilateration method [7]. The trilateration technique can be clearly shown in Figure 2.

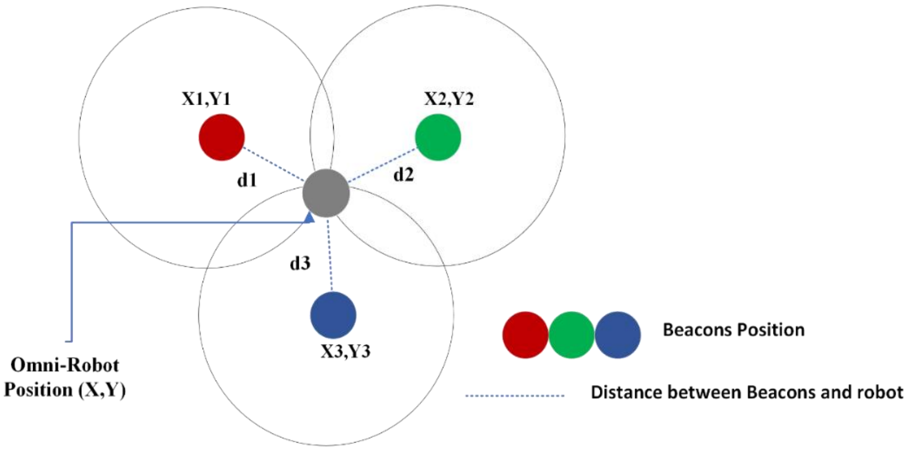


Figure 2. Trilateration technique.

Considering the basic formula for the general equation of a circle which is shown in equation (2)

$$d^2 = X^2 + Y^2 \tag{2}$$

where d is the radius of the circle, the equations of three circles centered at points (X_1 ,Y_1), (X_2 ,Y_2) and (X_3,Y_3) can be formed as shown in equations (3), (4), and (5) respectively:

$$d_1^2 = (X - X_1)^2 + (Y - Y_1)^2 \tag{3}$$

$$d_2^2 = (X - X_2)^2 + (Y - Y_2)^2 \tag{4}$$

$$d_3^2 = (X - X_3)^2 + (Y - Y_3)^2 \tag{5}$$

The distances d_1, d_2, and d_3 are known from the depth information of the camera. After the detection of the beacons, their coordinates (X_1, Y_1), (X_2, Y_2) and (X_3, Y_3) with respect to the global frame are known, which are set by the designer. Therefore, the two unknown variables (X, Y) which represents the point of intersection of the three circles and the robot current position can be determined by solving the equations (3), (4), and (5) as follows:
After subtracting equation (4) from (3) we obtain:

$$-2XX_1 + 2XX_2 + {X_1}^2 - {X_2}^2 - 2YY_1 + 2YY_2 + {Y_1}^2 - {Y_2}^2 = d_1^2 - d_2^2$$

$$X(-2X_1 + 2X_2) + Y(-2Y_1 + 2Y_2) = d_1^2 - d_2^2 - {X_1}^2 + {X_2}^2 - {Y_1}^2 + {Y_2}^2$$

Let: $\mathrm{A} = -2X_1 + 2X_2$, $B = -2Y_1 + 2Y_2$ and $C = d_1^2 - d_2^2 - {X_1}^2 + {X_2}^2 - {Y_1}^2 + {Y_2}^2$

This yields a new equation:

$$AX + BY = C \tag{6}$$

And by subtracting equation (5) from (4) the following equations are formed:

$$X(-2X_2 + 2X_3) + Y(-2Y_2 + 2Y_3) = d_2^2 - d_3^2 - {X_2}^2 + {X_3}^2 - {Y_2}^2 + {Y_3}^2$$

Where: $D = -2X_2 + 2X_3$, $E = -2Y_2 + 2Y_3$, $F = d_2^2 - d_3^2 - {X_2}^2 + {X_3}^2 - {Y_2}^2 + {Y_3}^2$

A new equation is formed:

$$DX + EY = F \tag{7}$$

Then, by multiplying (6) with (E) and (7) with (B) the following equations have been obtained:

$$AEX + BEY = CE \tag{8}$$

$$DBX + EBY = FB \tag{9}$$

After subtracting (9) from (8) we obtain:

$$X = \frac{CE - FB}{AE - DB} \tag{10}$$

by multiplying (6) with (D) and (7) with (A) we obtain:

$$ADX + BDY = CD \tag{11}$$

$$DAX + EAY = FA \tag{12}$$

After subtracting (12) from (11) we obtain:

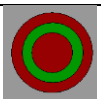
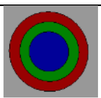
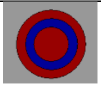
$$Y = \frac{CD-FA}{DB-AE} \tag{13}$$

The equations (10) and (13) are simply implemented in CoppeliaSim.

4. Artificial Passive Beacon Design

Practically, the vision sensor depends on the color's signature to detect the object. In the design of the beacon, two or more colors' signature is placed together to form a color code signature (CCs). The design is based on the combination of three disks with distinct sizes and colors (red, green, and blue only). The disks are placed on top of each other and have the same center. As a result, twelve beacons can be generated. Table 1 shows three designed shapes of the beacon and its codes. These artificial passive beacons are placed on the ceiling and used for indoor localization.

Table 1. The designed shape of the beacon.

Color Tags	Color Codes	Beacons
Red-Green-Red	121	
Red-Green-Blue	123	
Red-Blue-Red	131	

The designed beacon is placed on the ceiling with specific arranging that helps the vision sensor to detect at least three beacons at the same time. Figure 3 illustrates the arrangement of the designed beacons on the ceiling.

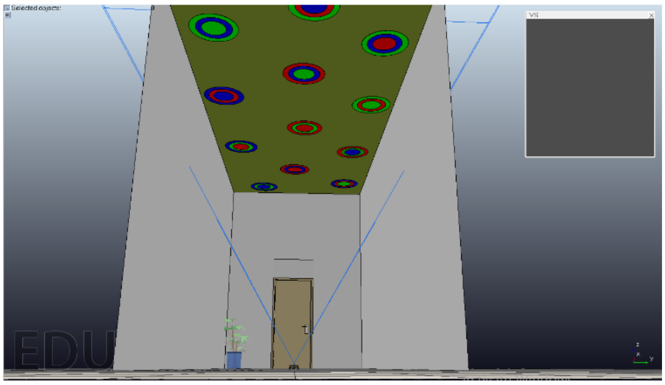


Figure 3. Beacons are set on the ceiling.

4.1. Color-coded beacon detection framework

The vision sensor in CoppeliaSim has built-in image processing filters. A combination of these filters is used to detect the designed tag. As shown in Figure 4, three filters with appropriate parameters are used to detect only the three primary colors (RGB) in the images and return binary images with ones in the pixels with red, green, and blue colors and zero otherwise. After that, three filters are used to segment the images returned from the first three filters. The goal of segmentation is to find the pixels with values of ones that are close to each other, Binary Large

Object (BLOB)[15]. These filters eliminate the BLOBs with small sizes and also return the coordinates of the center of each blob in the image coordinates in pixels. Each color code signature consists of three blobs of two or three different colors that have the same center coordinates in the image coordinates.

The detection of the CCs beacon is as follows: Frist, the distance between a blob center of a certain color and every other blob center of the same and different color is calculated using the distance formula between two points. If this distance is less than a certain threshold then the three blobs belong to the same beacon. The process is repeated until all tags are detected and recorded. Second, a color code is assigned to each beacon based on the sizes of the blobs with the assumption that red = 1, green = 2, and blue = 3, see Table 1.

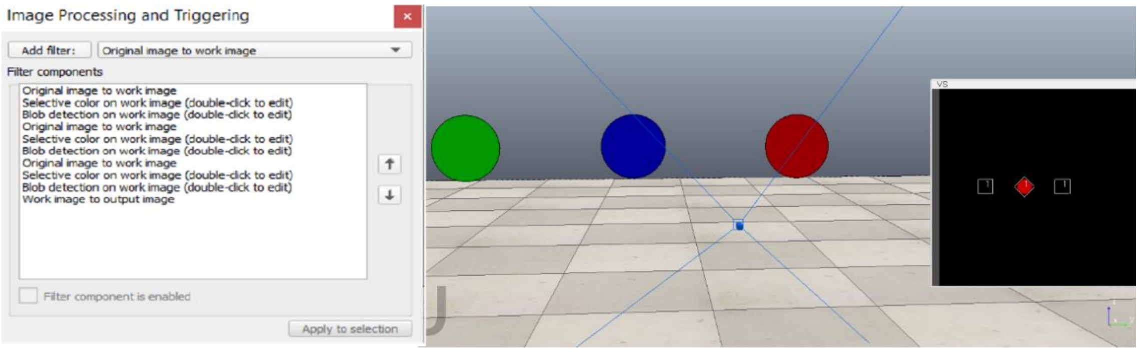


Image Processing and Triggering

Add filter: Original image to work image

Filter components

Original image to work image
Selective color on work image (double-click to edit)
Blob detection on work image (double-click to edit)
Original image to work image
Selective color on work image (double-click to edit)
Blob detection on work image (double-click to edit)
Original image to work image
Selective color on work image (double-click to edit)
Blob detection on work image (double-click to edit)
Work image to output image

□ Filter component is enabled

Figure 4. Blob detection of three colors (Red-Green-Blue).

5. Results and Discussion

The localization of the Omni-directional robot has been implemented using the simulation environment of CoppeliaSim which is controlled via scripts written in Lua language. Twelve circular objects with RGB color are used as beacons with a predefined position relative to the global frame. The scene in CoppeliaSim has a corridor with dimensions (3x8x4) in meters. The Omni-wheels robot designed in section (2) is used. The vision sensor is located on the top of the robot with a 60˚ FOV to detect at least three beacons at the same time. Two scenarios are considered to show the effectiveness of the proposed localization method.

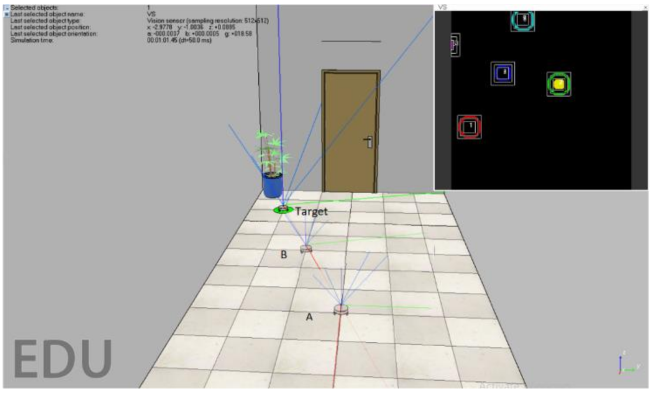
The first scenario is shown in Figure 5. The robot is located at point **A,** and the target is located at pose ($x_t = -3$ m, $y_t = 1$ m,$\theta_T = -11.6$˚). In the first stage, the robot at point **A** starts rotating using the following control law :

$$\dot{\theta} = k_{p\theta}(\theta - \theta_d) \tag{14}$$

where θ is the current robot orientation, $\theta_d = atan2(y_t - y_r, x_t - x_r)$, (y_r, x_r) and (y_t, x_t) is the Cartesian coordinates for the robot and target positions respectively and the controller parameter $k_{p\theta} > 0$. In the second stage, trilateration equations (10) and (13) are applied to obtain the robot's position relative to the global frame. The robot is then moved with a velocity $\dot{X}_R$that is calculated according to the equation:

$$\dot{X}_R = K_P\, \rho\, \cos\alpha \tag{15}$$

where $\dot{X}_R$ is the desired speed of the robot (with the assumption of $\dot{Y}_R = 0$), and controller parameter $K_P > 0, \rho = \sqrt{(y_t - y_r)^2 + (x_t - x_r)^2}$ and $\alpha = 0$ due to the previous rotation by θ_d. Finally, the robot is rotated to achieve the desired orientation θ_T using the control law (14) with $\theta_T = \theta_d$ The results are shown in Table 2.



Selected objects: 1
Last selected object name: VS
Last selected object type: Vision sensor (sampling resolution: 512x512)
Last selected object position: x: -2.9778 y: -1.0036 z: +0.0885
Last selected object orientation: a: -000.0007 b: +000.0005 g: +018.58
Simulation time: 00:01:01.45 (dt=50.0 ms)

Figure 5. The first scenario of the robot localization.

Table 2. The result of scenario 1.

	X_{Actual} (m) (Robot)	X_{Tri} (m) (Robot)	Y_{Actual} (m) (Robot)	Y_{Tri} (m) (Robot)	Distance error (m)= Target position – Robot position
A	0	0.08	0	0.02	3.16
B	1.227	1.209	0.413	0.404	1.88
Target	3	2.95	1	0.99	0.05

Figure 6 shows that The Omni-Wheels robot starts rotating from its initial orientation until reaching θ_d with an execution time of 4 seconds.

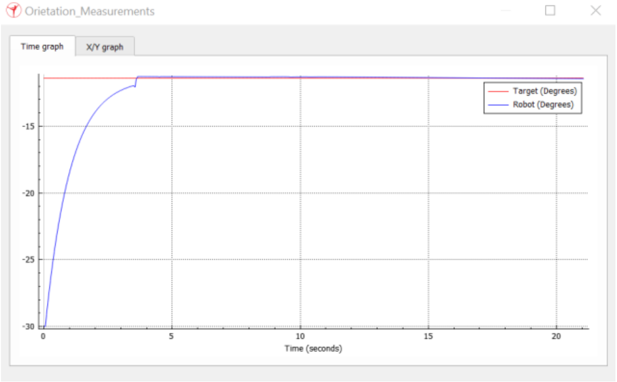


Figure 6. The orientation measurements.

Additionally, the (X, Y) coordinates of the robot from the initial position until reaching the target are shown in Figure 7. The graph shows that the robot coordinates do not change for the first 4 sec due to the rotation of the robot; after that, it can be seen that the robot (X, Y) coordinates change exponentially until reaching the target's coordinates.

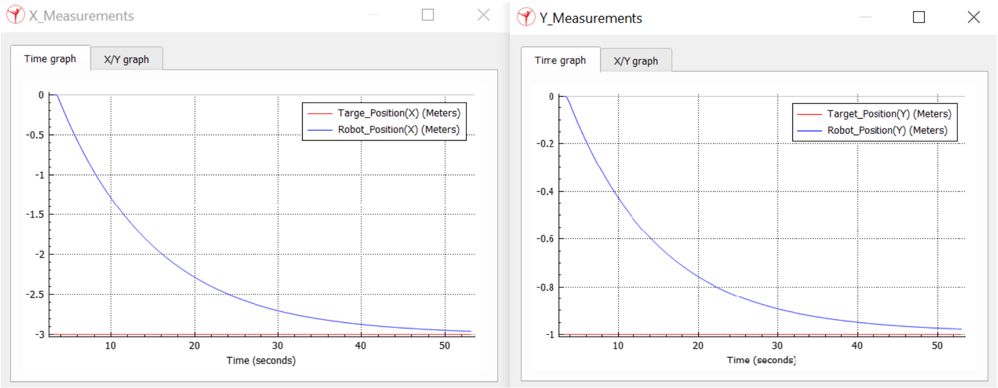


Figure 7. The Omni-wheels robot coordinates.

In the second scenario, as shown in Figure 8, the target is located at the pose ($x_t = 2$ m, $y_t = 1$ m, $\theta_T = 176^\circ$). The robot starts rotating at point **A** and then the steps of the first scenario are repeated. The robot reaches the target as demonstrated in Table 3.

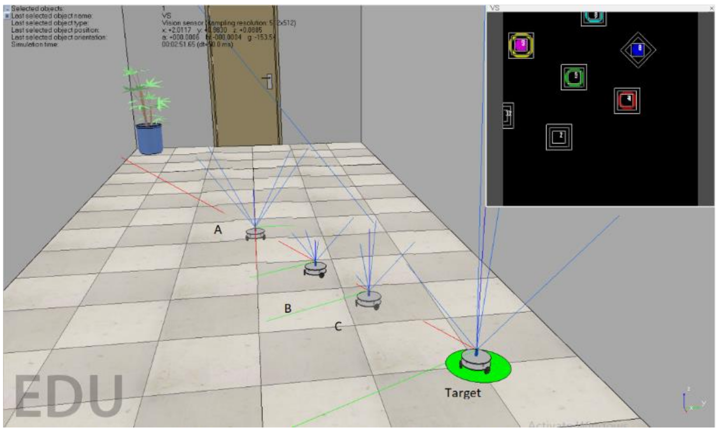


Figure 8. The second scenario of the robot localization.

Table 3. The result of scenario 2.

	X_{Actua}(m) (Robot)	X_{Tri}(m) (Robot)	Y_{Actual}(m) (Robot)	Y_{Tri}(m) (Robot)	Distance error (m)= Target position – Robot position
A	0	0.039	0	0.019	2.192
B	0.693	0.673	0.339	0.329	1.487
C	1.295	1.288	0.637	0.633	0.801
Target	2	1.991	1	0.978	0.023

Figure 9 shows that The Omni-Wheels robot starts rotating from its initial orientation until reaching θ_d with an execution time of 5 seconds.

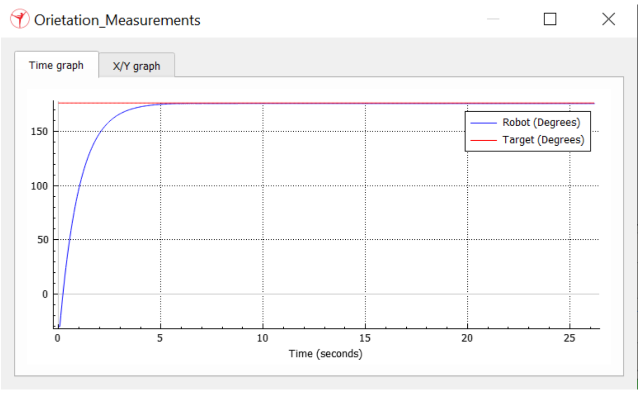


Figure 9. The orientation measurements.

Additionally, the (X, Y) coordinates of the robot from the initial position until reaching the target are shown in Figure 10. The graph shows that the robot coordinates do not change for the first 5 seconds due to the rotation of the robot; after this time, it can be seen that the robot (X, Y) coordinates change exponentially until reaching the target's coordinates.

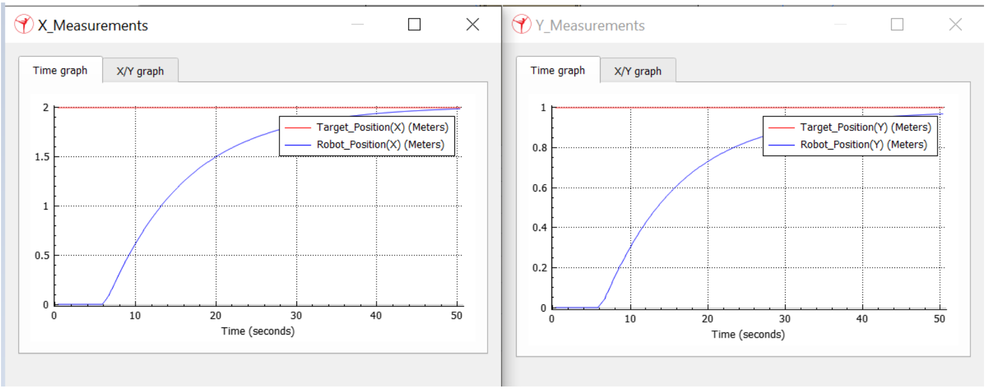


Figure 10. The Omni-wheels robot coordinates.

6. Conclusion

In this paper, the indoor localization of the Omni-wheels robot was simulated using the CoppeliaSim robot simulator. The indoor localization of the robot was achieved by applying the trilateration method. This method depends on identifying color-coded beacons to obtain their position with respect to the global frame and calculating the distances between each beacon and the robot. These beacons have been designed in circular shapes so that the detection of them can be done easily from any robot orientation. Two scenarios were examined to prove the capability of the proposed method for robot indoor localization. In both scenarios, the simulation results show that the robot rotates around itself to achieve the desired angle, then moves towards the target pose. The designed CCs beacons have enabled the Omni-wheels robot to estimate its position and reach the desired target.

References

[1] Fabregas, E., Farias, G., Peralta, E., Vargas, H. & Dormido, S. Teaching control in mobile robotics with V-REP and a Khepera IV library. In 2016 IEEE conference on control applications.2016. IEEE. pp.821-826.

[2] Francisco R., Francisco V. and Carlos L.A.,. A review of mobile robots: Concepts, methods, theoretical framework, and applications. International Journal of Advanced Robotic Systems.2019. 16(2).1-22

[3] Pablo, C., Miguel, V., David, C., Luis, M. and Manuel, B., Indoor Robot Positioning using an Enhanced Trilateration Algorithm. Engineering and Industrial Development Center (CIDESI).2016. 13(3).1-8

[4] Chuang, L. Object localisation strategy for a mobile robot using RFID.2012. 1(1). 5.

[5] Gueaieb, W., and Miah, M.S. An intelligent mobile robot navigation technique using RFID technology. IEEE Transactions on Instrumentation and Measurement.2008. 57(9), pp. 1908-1917.

[6] Mohanad, N. N., Mohammed, Q., and Omar, Y. I. Landmarks exploration algorithm for mobile robot indoor localization using vision sensor. Journal of Engineering Science and Technology. 2021. 16(4). pp. 3165 – 3184.

[7] Ladd, A.M., Bekris, K.E., Rudys, A.P., Wallach, D.S. and Kavraki, L.E. On the feasibility of using wireless ethernet for indoor localisation. IEEE Transactions on Robotics and Automation.2004. 20(3), pp. 555-559.

[8] Felipe, P. C., Joao, A. F., and Andre S. O. Integrating mobile robot navigation control, visual object detection and manipulation using ROS and V-REP. Conference: Congresso Brasileiro de Inteligência Computacional.2020.134(10).5.

[9] Bittel, O. and Blaich, M. Mobile robot localisation using beacons and the Kalman filter technique for the Eurobot competition. International Conference on Research and Education in Robotics. Springer.2011.pp. 55-67.

[10] Huosheng, H., and Dongbing, G. Landmark-based navigation of industrial mobile robots. Industrial Robot- an International Journal.2000.27 (6). pp. 458-467.

[11] Bekris, K.E., Argyros, A.A. and Kavraki, L.E. Angle-based methods for mobile robot navigation: Reaching the entire plane, Robotics and Automation. IEEE International Conference on 2004, pp. 2373-2378.

[12] Huosheng, H., and Dongbing, G. Landmark-based navigation of industrial mobile robots. Industrial Robot: An International Journal.2000. 27(6), pp. 458-467.

[13] Kostomins, A., and Osadcuks, V. Infrared active beacon application evaluation for mobile robot localisation in agriculture. Engineering for Rural Development (Latvia).2014.pp. 373-378.

[14] Hugh, D.,W.Where am I? A tutorial on mobile vehicle localization. Industrial Robot: An International Journal.1994, 21(2). pp. 11-16.

[15] Alexander, B., Rainer, H., and Kenneth, B.K. Hardware Acceleration of BLOB Detection for Image Processing.Third International Conference on Advances in Circuits, Electronics and Micro-electronics.IEEE.2010.pp 28-33.