

# Automatic Navigation of Mobile Robot Based on Wheel's Encoder and Predefined Map

Hala Jamal Hadi <sup>1</sup>, Prof. Dr. Salih Al-Qaraawi <sup>2</sup>

<sup>1</sup>Computer Engineering Department, University of Technology, Iraq

<sup>2</sup>Control and System Engineering Department, University of Technology, Iraq

<sup>1</sup>halajhadi94@gmail.com, <sup>2</sup>drsalihalqaraawi@gmail.com

**ABSTRACT**— Recently, the automatic movement of mobile robots has played a very important role in the advancement of technology. Automated mobile robot path determination is one of the most important challenges in the science of technology. This paper proposed a path planning method for wheeled mobile robots based on a real time calculation of a predefined distance on a certain map to enable the mobile robot to navigate at indoor areas according to the calculated distances and angles on the paths. The proposed system uses two wheels' car as a prototype with two optical encoders to determine the number of wheel's rotations, in order to calculate the needed distances and angles between two points on the navigation path. The system was controlled by a microcomputer Raspberry Pi, programmed using python programming language. The experimental results show an accurate distances and angles measurement for the navigation under a suitable condition.

**Keywords:** Automatic Movement, Mobile Robot, Path Tracking, Raspberry PI, Optical Encoder, Python.

## I. INTRODUCTION

An autonomous mobile robot is a programmable and multitasks mechanical device that can be employed in many different applications. Mobile robot's navigation can be accomplished by the way of using many 'ready to use' algorithms that can easily facilitate its movement from one point to another [1]. The term "navigation" is defined as the process of monitoring, controlling, and updating the movement of the object from starting point of movement to the destination point along the route [2].

Mobile robot can be remotely operated or autonomously; the former is used to access hazardous or inaccessible places hostile to people, allowing a human operator to determine and control the robot's movements, while the automated robot can take its decision of movement according to the environment that work on and tasks that it used for which is independently of human intervention. Such that, there is outdoor and indoor environments. The navigation at outdoor may differ from indoor because of the limitation of using GPS at indoor systems.

It is also necessary to take into consideration the path that the robot should follow from point to another to take its decision to move. A number of researches have been proposed different navigation methods for different mobile robot applications for example; in 2012, Mustafa Engin and Dilad Engin [3], presented line follower robot that follow a drawn line on the floor using eight infrared sensors, since it detected the black color line it will move along it a navigation path. In the next year, Jeisung Lee et al [4], presented an approach for AGV that is also based on the marker recognition on the floor using probabilistic approach by using distance transform matching. Furthermore, in 2015, a snow removal robot's navigation was accomplished according to marker detection by a low resolution camera, the markers were placed at each segment of the snow, and the robot should remove these snows based on the placed markers, the problem here is that the vision system needs a high processing time and good resolution camera to obtain accurate result [5]. In 2018, authors in [6], designed a CIoT medicine robot

Received 7 April 2019; Accepted 5 September 2019

delivery system, based on multicore embedded system and it has been based on cloud and IoTs technology. At the same year, Tyrone Sherwin et al. [7], proposed emitter-based localization and navigation robot system based on single RF emitter and A\* path planning technique. Also in 2018, Diksha Singh and Anil Nandgaonkar [8] proposed a surveillance robot system that used a camera, to control the direction of robot's movement by the webpage via Wi-Fi connection.

In this paper, the problem of robot navigation for indoor application has been addressed and the path that the robot should follow depending on the distance between two points taking into consideration the angles of turns. In order to achieve this goal, the distance should be measured accurately to get accurate result. In other words, the path of mobile robot is calculated using distance that travelled by robot which is calculated from the fact of how many rotations the robot's wheels do until it completes the given distance to reach the destination. Rotations counting can be achieved by counting the ticks of wheel's rotation using optical shaft encoder.

## II. THE PROPOSED SYSTEM DESIGN

### A. Hardware System Equipment

#### i. Raspberry PI 3B

The Raspberry PI model 3B is a microcomputer with powerful features. It has been built-in Wi-Fi, which can be controlled and accessed remotely via SSH or any other method from PC. The Raspberry PI has a built-in Linux system with python IDE which is a powerful programming language with a lot of libraries. This microcomputer generates the appropriate signal or command to be sent to the car to move in a specific direction and for the needed distance, or to stop the car if the needed distance has been finished.

#### ii. Shaft Encoder

It is an optical encoder that is attached to each wheel of the car and it is connected directly to the microcomputer. It is used to calculate the steps moved by the car in real time. It consists of two components; the IR optical sensor which contains a light transmitter and receiver, as well as, the circular disk which contains a number of holes, the resolution of the encoder increases as the number of holes' increase. Once the car starts to move, the microcomputer starts to calculate the steps which consequently represent the number of rotations of the wheel, i.e., the distance moved. The working principle of the shaft encoder is as follows:

1. A light transmitter is positioned on one side of the disk, and the light detector is positioned on the other side of the disk.
2. As a disk is rotating the output of the receiver side switch between ON/OFF state, due to the transparent and opaque slots of the disk which is one for reflecting the light and the other is for passing the light.
3. When the light goes through the transparent slot, it will be reflected and indicating one countering state.
4. When the counter becomes equivalent to the number of the holes, it refers to one revolution of the disk. Therefore, the disk has rotate one rotation which represents the circumference of the disk. Figure 1 shows the working operation of the optical encoder.

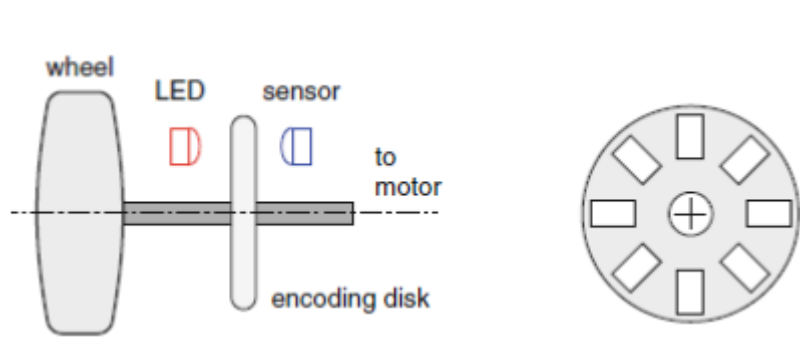


FIG. 1: THE WORKING OPERATION OF OPTICAL ENCODER [9].

### iii. L298n Motor Driver

It is an electronic module that controls the direction of the car by controlling the rotation direction of the DC motor(s), it has 4 connectors to be connected to the microcomputer. Another task is to control the speed of the DC motors using PWM Technique. Each DC motor is connected to L298N H-Bridge motor driver which controls the speed and directions. The motor driver is connected to microcomputer Raspberry PI 3B. Two optical encoders are attached to the wheels to calculate number of steps (ticks) by counting the holes that included in a circular disk, i.e. number of rotations, which is consequently related to the distance moved by the car.

The system block diagram is represented in Figure 2:

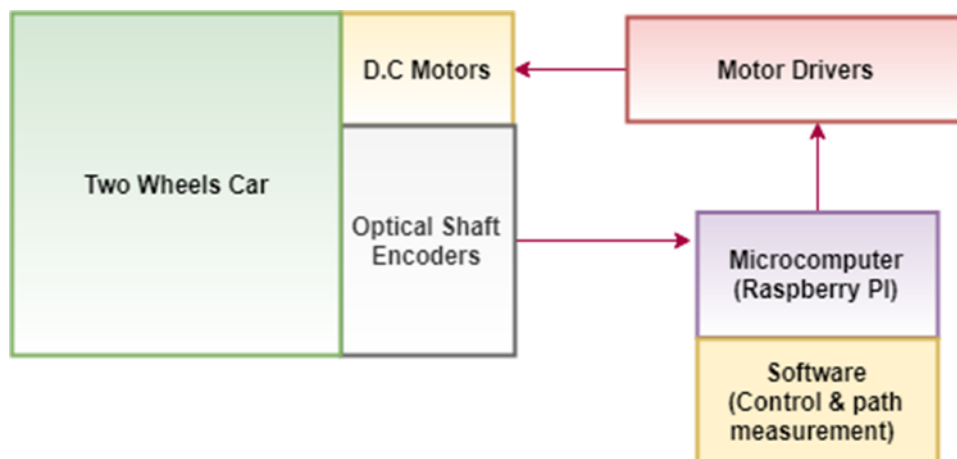


FIGURE 2: SYSTEM BLOCK DIAGRAM.

### B. The proposed system representation and path measurements:

In this proposal, a two wheels' robotic car is used as a prototype for robot representation to automatically moving in a small map that contains some straight lines and curves. Basically, a method for robot navigation based on pre-measured distance(s) on virtual lines of movement has been proposed. The first method was based on measuring the needed time to move for a specific distance using the relationship between the distance and time, where the distance = time \* speed, and it was not accurate and a lot of errors was introduced due to power's limitation. So that, the other proposed method to the calculated distance doesn't depend on the power level of the system, which limits the stability of the car's speed. However, the system uses two optical encoders attached to each wheel of the car to count the number of wheel's rotation and thus finding the needed distance and angles. The distance can be obtained using the following formula:

Received 7 April 2019; Accepted 5 September 2019

$$\text{Distance to be moved} : \text{Wheel's Circumference} * \text{Number of Rotations} \quad (1)$$

Number of rotations can be obtained depending on the number of holes of the encoders used in the system. Thus the size of the wheels as well as the holes' number are the most important factors in the distance calculation, changing their values will affect the accuracy of distance measurement. In the proposed system an optical encoder with 20 holes is used, with wheels' circumference of 20 cm. Thus one rotation is equivalent to 20 holes counted by the encoder which in turn equal to 20 cm moved by the car. To calculate the number of rotations of the encoder, the following formula is used:

$$\text{No. of rotations} = \frac{\text{No. of ticks counted by encoder}}{\text{No. of ticks for one revolution}} \quad (2)$$

Where the "tick" indicates the counted hole in a circular disk that attached to the wheels. Which represents the other part of the encoder. Now, to rotate the robot car for a specific angle (specific direction), swing rotation phenomena is used as shown in Figure 3. The swing rotation simply based on stopping one wheel and running the other one, so that the direction of rotation will be toward the stopped wheel. A circle around the stopped wheel is created so that the stopped wheel be the center of this circle. To measure the distance that one wheel should move to rotate the robot for a specific angle, the angle should be known so that the number of ticks (steps) of movement, which is fraction of one wheel's rotation, could be calculated. The distance between the middle of two wheels represents the radius of the circle that the robot will draw for the turning. To turn 360 degrees, the one wheel should complete one rotation around the other fixed wheel which means that this distance is equivalent to the circumference of the created circle. In other words,  $(Ci = 2 \pi Ri)$ , where,  $Ci$  represents the circumference of the created circle around the stopped wheel. i.e. the arc. Additionally,  $Ri$  represents the radius of this circle.

If one wheel of the car moves a distance equal to  $Ci$ , it creates a  $360^\circ$  turning angle. Thus, the angle of rotation could be obtained from the arc of the circle created by the moved wheel. In order to calculate the arc that the car should move to create a specific angle, the basic mathematic science is used [9]:

$$\frac{\text{Arc}}{\theta} = \frac{Ci}{360^\circ}$$

or:

$$\text{Arc} = Ci * \frac{\theta}{360^\circ} \quad (3)$$

Table 1 represents the relationship between angle of turning, distance to be moved for turning and number of ticks based on the formula (3) For example, if  $90^\circ$  turn is needed, 90 can be divided by 360 and multiply it by the circumference of the created circle which equivalent to 85 cm. The proposed system algorithm is represented in Figure 3.

TABLE 1: THE RELATIONSHIP BETWEEN THE ANGLES OF THE CHANGING DIRECTION AND THE NUMBER OF ENCODER'S STEPS.

Required Turning Angle $\theta$	<i>distance</i> on circumference for specified $\theta$	Distance on Arc (cm)	No. of ticks (Steps)
360°	$(2 \pi Ri) * 360/360$	84.82	85
180°	$(2 \pi Ri) * 180/360$	42.41	42
120°	$(2 \pi Ri) * 120/360$	28.27	28
90°	$(2 \pi Ri) * 90/360$	21.20	21
72°	$(2 \pi Ri) * 72/360$	16.97	17
60°	$(2 \pi Ri) * 60/360$	14.13	14
51°	$(2 \pi Ri) * 51/360$	12.11	12
45°	$(2 \pi Ri) * 180/360/8$	10.60	10
38°	$((2 \pi Ri)38/360$	8.95	9
30°	$(2 \pi Ri) * 30/360/12)$	7.06	7
18°	$(2 \pi Ri) * 18/360$	4.20	4
12°	$(2 \pi Ri) * 12/360$	2.82	3

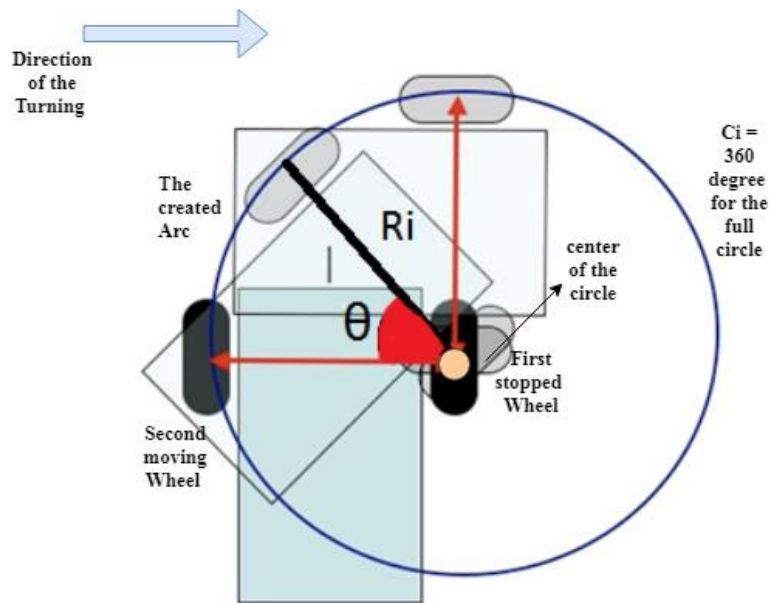


FIG. 3: SWING ROTATION.

### The Proposed System Algorithm

The proposed system algorithm which is applied, consists of the following steps:

1. Obtain the map of the building.
2. Specify the paths on the map
3. Specify the distances and angles of each path
4. Count the distance in term of ticks (no. of holes by encoder), use counter  $c$
5. Run the car, Initialize counter  $i=0$ ,
6. When the car moves, increase  $i$ ,  $i=i+1$ .

Received 7 April 2019; Accepted 5 September 2019

7. If there are any turnings, the car will turn to a specific direction for a specific angle by stopping one wheel and run the other one depending on the direction of movement. The moving wheel starts to count the number of ticks that it should move to create a specified angle.
8. If  $i=c$ , then, stop the car.

The general flowchart of the proposed system is shown in Figure 3.

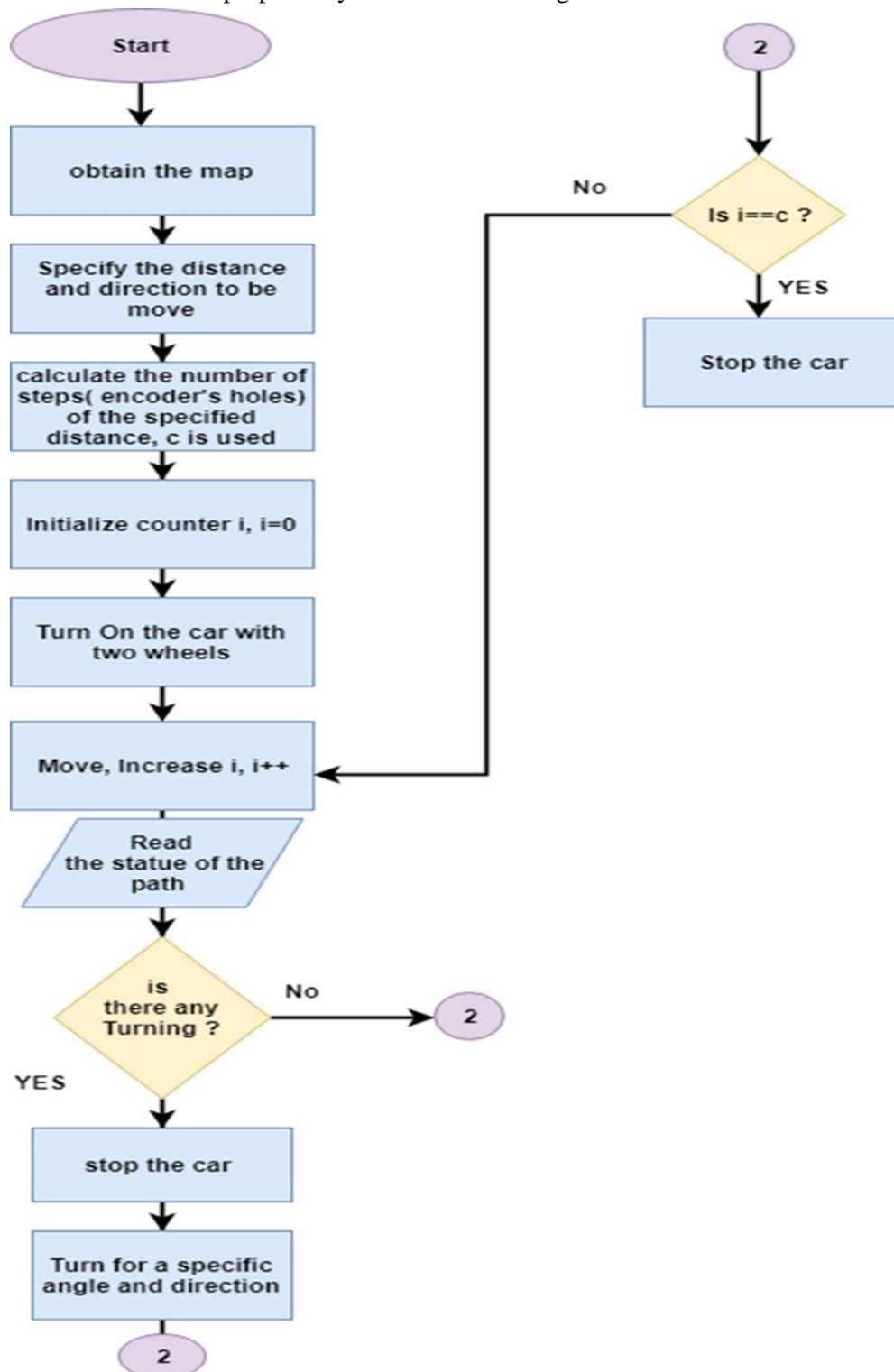


FIG.4: FLOWCHART OF THE PROPOSED SYSTEM.

Received 7 April 2019; Accepted 5 September 2019

### C. TESTING OF THE PROPOSED SYSTEM

To test the proposed system, two different scenarios were considered to check the accuracy of using encoders on different floors. The first one was about moving the car on a specific path on a flat surface, while the other one was about moving the car on a carpet. The first scenario considered two cases:

1. The direct movement of the robot (without turnings) for a specific path between two points which is represented in Figure 5. The robot car moved with 300 steps counting by encoders (3 meters) in the forward direction.



FIG. 5: THE DIRECT MOVEMENT OF THE CAR WITHOUT TURNINGS.

2. The second case was considered with turning angle of rotation as 90 degrees, the angle is chosen based on the planning path on the map, the robot moved 5m (500 steps) then it stopped and turned 90 degrees to the left, i.e., the right wheel rotates 21 steps. After that it moved 6m (600 steps) then stopped. It turned to right 90 degrees and completed its movement for 2m (200 steps) to reach the destination point. Figure 6 represents the path that the robot should track from the starting point P1 until the destination point P4 with two 90 degrees turning. Figure 6 represents the practical robot movement, whereas Figure 7-a shows the first point that the robot should move from until it reaches the target, Figure 7-g.

Table 2 represents the path calculation of moving the car on a path of Figure 5.

TABLE 2: PATH CALCULATIONS OF SCENARIO 2 OF THE PROPOSED SYSTEM

Points	Command	Distance/ Angle	Calculations	No. of Rotations and Steps
<b>P1-P2</b>	Forward	5 meters	500/20 cm	25 times = 500 steps
<b>P2</b>	Turn left	90°	$2\pi R_i/4$	1 rotation and 1 step = 21 steps
<b>P2-P3</b>	Forward	6 meters	600/20cm	30 times = 600 steps
<b>P3</b>	Turn right	90°	$2\pi R_i/4$	1 rotation and 1step = 21 steps
<b>P3-P4</b>	Forward	2 meters	200/20	10 times = 200 steps

Received 7 April 2019; Accepted 5 September 2019



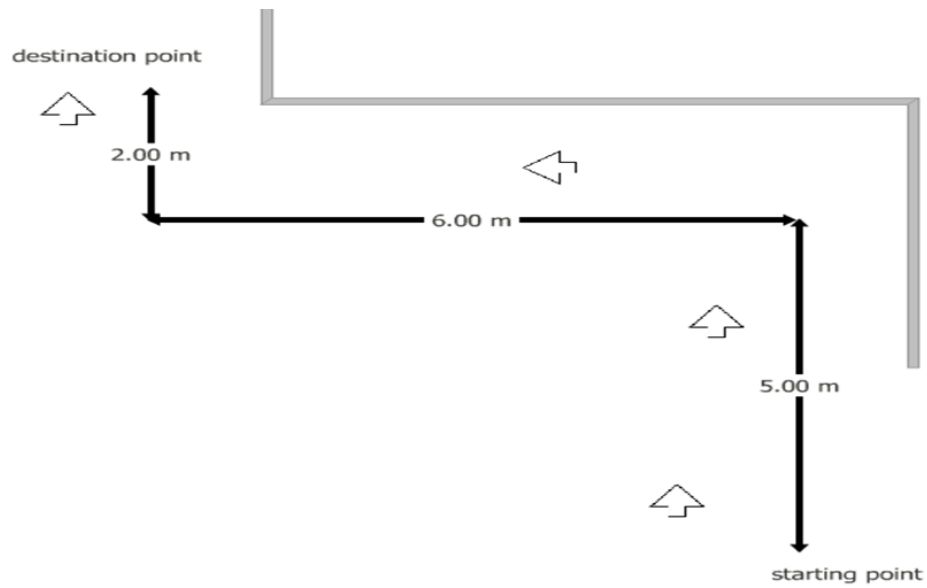
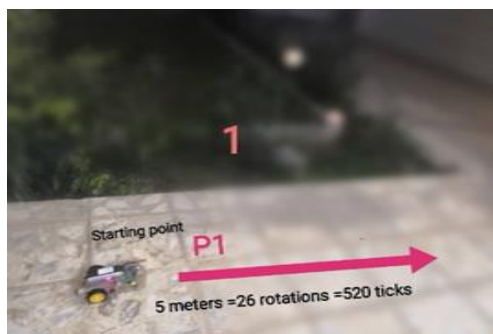


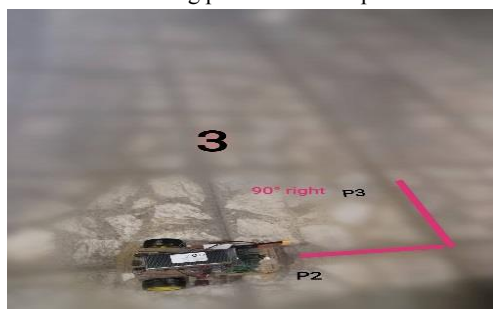
FIG 6: THE MAP OF THE SECOND SCENARIO OF THE PROPOSED SYSTEM.



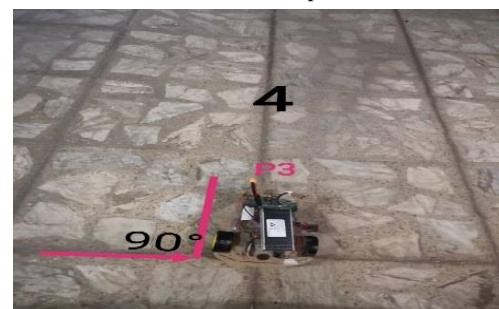
a: starting point of robot's path



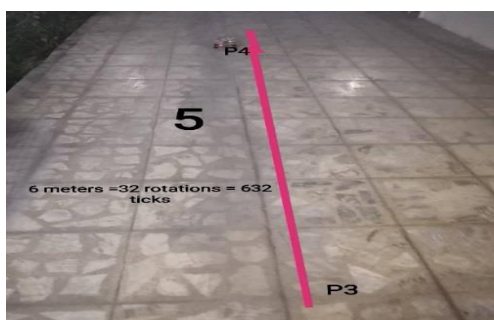
b: robot at middle point P2



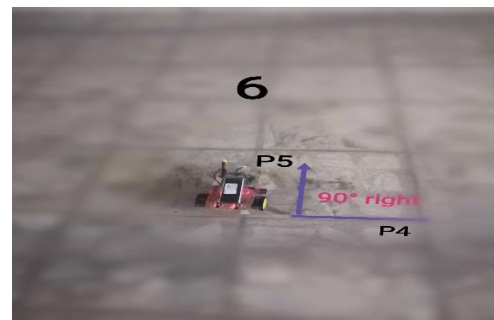
c: Robot turning 90-degree to the right



d: robot at P3 after turning



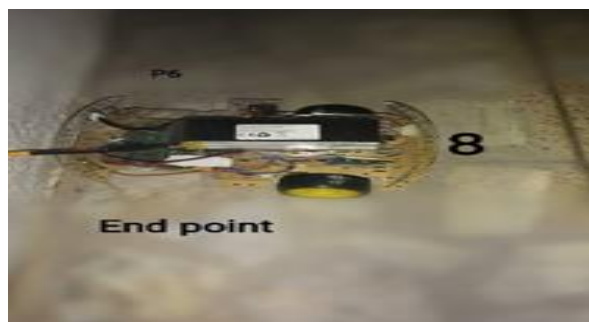
e: robot at point 4



f: robot at p5 after turning

Received 7 April 2019; Accepted 5 September 2019





g: The Robot at the target position

FIG. 7: EXPERIMENTAL TEST OF THE PROPOSED SYSTEM.

The second scenario tested the movement of the car on the carpet. Figure 8 shows the practical experiment of moving the car on the carpet.

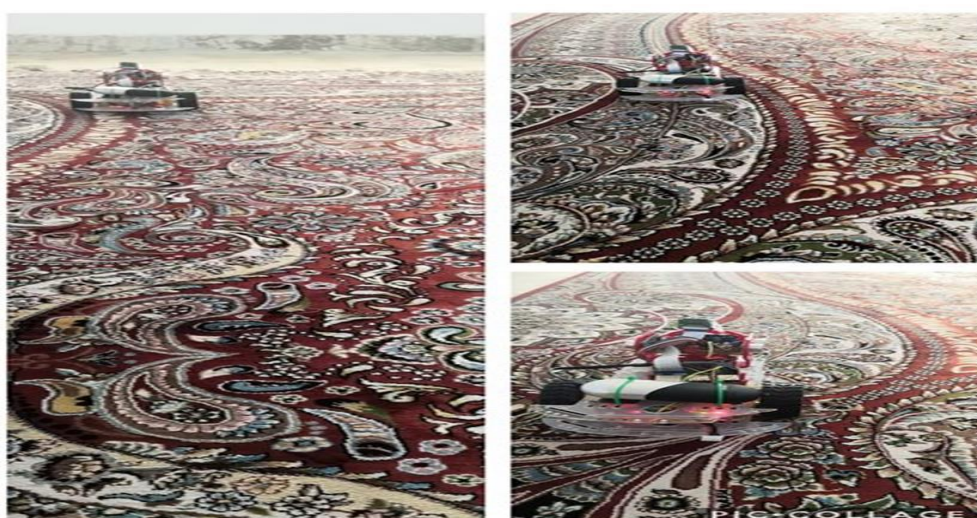


FIG. 8: THE SECOND SCENARIO OF MOVEMENT THE CAR ON THE CARPET.

The accuracy of the distance measurement using optical encoders depends mainly on the resolution of the used encoder and the surface that the car moves on. In the first scenario the car moved on a flat surface and the expected results were obtained. But when the car is moving on a rough surface, some errors have been occurred due to the instability of the surface which has negative effects on encoder counting. As a result, the accuracy was 100% when moving the car on the flat surface and 92% when it is moving on a carpet. It is necessary to mention that the accuracy here indicates the number of success attempts of movement over the whole attempts \*100%. A comparison between the proposed system and the system used in reference [5] is demonstrated in Table 3.

TABLE 3. A COMPARISON BETWEEN THE PROPOSED SYSTEM AND THE SYSTEM IN REFERENCE 5.

Criteria	The proposed system	System in reference 5
Technique	Path measurement	Line tracking
The path	Virtually drawn	Physically drawn
Cost	Less	More
Power consumption	Less	More
Weather condition	Doesn't affected	Affected

Received 7 April 2019; Accepted 5 September 2019

### III. CONCLUSION

This paper presents a proposed path tracking a method between two points in a small area. The proposed system provides the ability to many applications that depend on the mobility of the robot, like surveillance, delivery, detection, guided and other systems. An electric car that controlled using Raspberry PI microcomputer was used in the system. Moreover, an optical encoder was used to determine the number of wheels' rotations, in order to calculate a needed distance between any two points. Turning of the car was also considered and the results show that the tracking is near the actual path. The system proved its reliability and accuracy as well as the low cost components used comparing to other system.

This work can be extended so as network connection should be established between the robot and another network device and the status of the robot movement will be monitored, as well as the consideration of destination point reaching notification.

### REFERENCES

- [1] Md. Arafat Hossain, Israt Ferdous, "Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique" *Journal of Robotics and Autonomous Systems*, vol. 64 (2015), pp. 137–141.
- [2] Claudio Urrea · José Muñoz," Path Tracking of Mobile Robot in Crops", *Journal of Intelligent Robotic Systems* vol. 80, no. 2, 2015, pp. 193–205
- [3] Ernesto Rivas, Koutarou Komagome, Kazuhisa Mitobe and Genci Capi," Image-Based Navigation for the SnowEater Robot Using a Low-Resolution USB Camera", *Journal of Robotics*, vol. 4, no. 2, 2015, pp.120-140
- [4] Jeisung Lee, Chang-Ho Hyun and Mignon Park," A Vision-Based Automated Guided Vehicle System with Marker Recognition for Indoor Use", *Journal of Sensors*, vol. 13, no. 8, 2013, pp. 10052-10073
- [5] Mustafa Engin, Dilsad Engin, "Path Planning of Line Follower Robot", *Proceedings of the 5th European DSP Education and Research Conference (EDERC)*, 2012, pp. 1–5.
- [6] Huiru Cao, Xiaofeng Huang, Jianyi Zhuang, Jianqiang Xu, Zening Shao, "CIoT Assisted Indoor Robot for Medicine Delivery", *Joint International Advanced Engineering and Technology Research Conference 2018, Advances in Engineering Research (AER)*, vol. 137, pp. 85-89.
- [7] Tyrone Sherwin 1, Mikala Easte, Andrew Tzer-Yeu Chen, Kevin I-Kai Wang, and Wenbin Dai, "A Single RF Emitter-Based Indoor Navigation Method for Autonomous Service Robots", *Journal of Sensors*, vol. 18, 585, 2018.
- [8] Diksha Singh and Anil Nandgaonkar," IOT-Based Wi-Fi Surveillance Robot with Real-Time Audio and Video Streaming", *Advances in Intelligent Systems and Computing*, vol. 810, *Proceedings of ICCASP 2018*, pp. 639-647
- [9] Michele Bousquet," *Physics for Animators*", CRC Press, 2015.
- [10] Mordechai Ben-Ari and Francesco Mondada, "element of robot", Cham: Springer Nature.2018.