ISSN: 04532198
Volume 62, Issue 11, December, 2020

Optimal Mapping Trajectory for Autonomous Robot Waiter

Faikul Umam[1], Sri Wahyuni[2], Hairil Budiarto[3], Robi Maulana Putra[4]

Department of Mechatronic Engineering, University of Trunojoyo Madura, Indonesia[1,2,3,4]

ABSTRACT— In the midst of the Covid-19 pandemic, the world has been attacked at this time, food delivery robots are one of the solutions to reduce connectivity between people in restaurants. Droplet is the transmission of Covid through coughing, sneezing, talking and also breathing out of the mouth and nose. This research has developed a prototype of a waiter robot that can run autonomously without markers to replace the role of humans in delivering food. A waiter robot can pass a previously mapped path. Track mapping is formed by moving the robot manually throughout the work area. The robot can generate path nodes and robot coordinates (x, y) from rotary sensors both encoder and odometry. After all trajectories of the work area have been mapped, the robot can determine the shortest path using Dijkstra's algorithm. The optimization of the movement of the autonomous waiter robot is built using three omni-directional. The results obtained through a series of tests with Dijkstra are that it is able to work well in finding the destination node from the origin node even though there are changes in the work area and the smooth movement generated by the three omni-directional trajectories.

KEYWORDS: Autonomous Waiter Robot, Mapping Trajectory, Odometry, Shortest Path, Dijkstra.

1. INTRODUCTION

Indonesia has many types of regional specialties with different flavors and characteristics. This is very beneficial for the economy of the Indonesian people with the nation's cultural diversity towards prosperity by developing the food industry, such as cafes and restaurants. However, at the end of 2019, the world was affected by a catastrophic virus that spread rapidly to all corners of the world, this was no exception by Indonesia. This virus is called Covid-19. The largest transmission of Covid-19 is identified through droplets when coughing, sneezing, talking and even breathing out of the mouth or nose. Various business sectors were affected, hitting the international economy, including the culinary sector. The government has disseminated several preventive measures to the public to minimize the spread of the virus through the use of masks, washing hands frequently, providing hand sanitizers, spraying disinfectants, maintaining distance and avoiding crowds. In order to avoid crowds and maintain a distance in certain spaces such as cafes and restaurants, food delivery robots are one solution to reducing human connectivity [1]. As a substitute for the role of humans in delivering food, this research has developed a prototype waiter robot that is able to walk autonomously without markers [2], [3]. Robot is in charge of bringing food and drinks ordered by customers from one table to another [4]. Research on waiter robots generally focuses on discussing navigation [5], mapping, for example those generated from the SLAM (Simulation Localization and Mapping) algorithm, or the use of line follower model robots [6]. In the navigation process, the robot tracks a line with a predetermined path so that the robot reaches the specified target [7]. But the drawback is that it really depends on the thickness of the color of the line that is passed. And the problem is that the color of the line will fade if the robot cannot detect the trajectory, so the line tracking robot cannot reach the destination point. In addition to navigation, waiter robots must also provide a system or features that make it easier for customers to order food [8]. These features can be in the form of buttons and an LCD that displays a list of food menus and a food order button [9]. Autonomous waiter robots have been developed based on the mapping of the work area to add strengths and review the weaknesses of previous research. Tests are carried out with several different work areas or node changes on various track models. There are several chairs and tables where customers wait for orders. At first, the work area was mapped to get a reference for the robot to carry out its duties. Odometry is used to estimate the change in robot position over time by mapping the robot's position on the cartesian axis. The data generated from the mapping activity is the position of the robot in the form of a coordinate point (node) and the direction of the robot (heading). The data is then used as parameters for the robot's navigation. The shortest path from the origin table to the destination table is obtained from a series of paths that have been mapped by applying Dijkstra's algorithm. The movement to carry out the navigation will be charged to three omni-directional so that the robot moves more flexibly and smoothly.

2. The Waiter Robot

The prototype of an autonomous waiter robot as in Figure 1 which was developed in this study has supporting parameters, including: the robot has a size of 60 x 60 cm, and it has the ability to navigate, path planning and control systems. The part of the autonomous waiter robot consists of a tray, external rotary encoder, omni-directional sensor, omni-directional, motor and IMU sensor. An external rotary encoder is used for mapping the robot's path. In order for the robot movement to be smooth [10], this robot uses an omni-directional wheel [11], [12]. This robot has two types of Omni-directional wheels, (1) omni-directional wheel to move the rotary encoder sensor, which records all robot displacements, (2) omni-directional wheels as a robot wheel that will receive action from the motor.

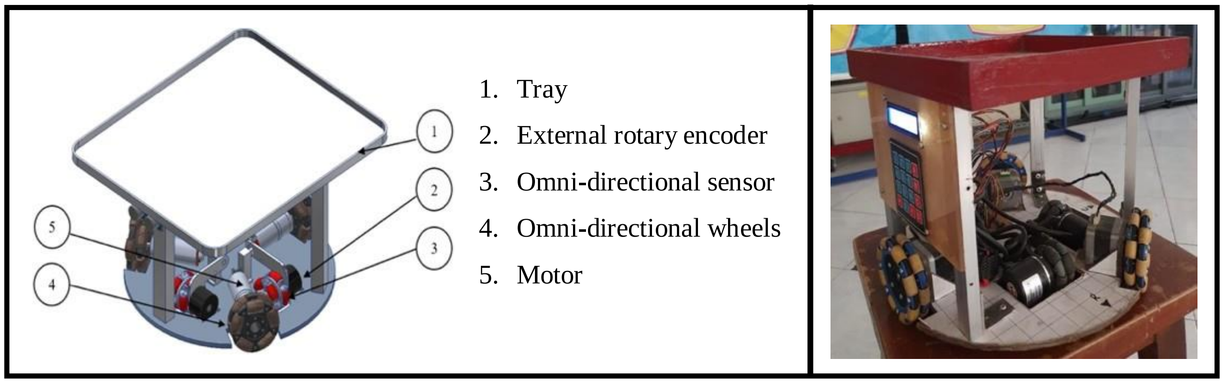
1
2
3
4
5
1. Tray
2. External rotary encoder
3. Omni-directional sensor
4. Omni-directional wheels
5. Motor

Figure 1. Design of Autonomous Waiter Robot

This robot has a mapping mode and a serve mode. If the work area is new and unmapped, the user must first map the area by selecting the mapping mode. Area mapping is done by manually pushing the robot over the entire work area trajectory until it returns to the starting point. Each displacement, the change in the position value of the robot is known using a rotary encoder sensor which is processed by odometry. If the work area mapping process is at the intersection point, the user must press the node button, and if the robot arrives at the customer table, the user must press the table button. This path data is then converted into a graph consisting of edges and nodes.

ISSN: 04532198
Volume 62, Issue 11, December, 2020

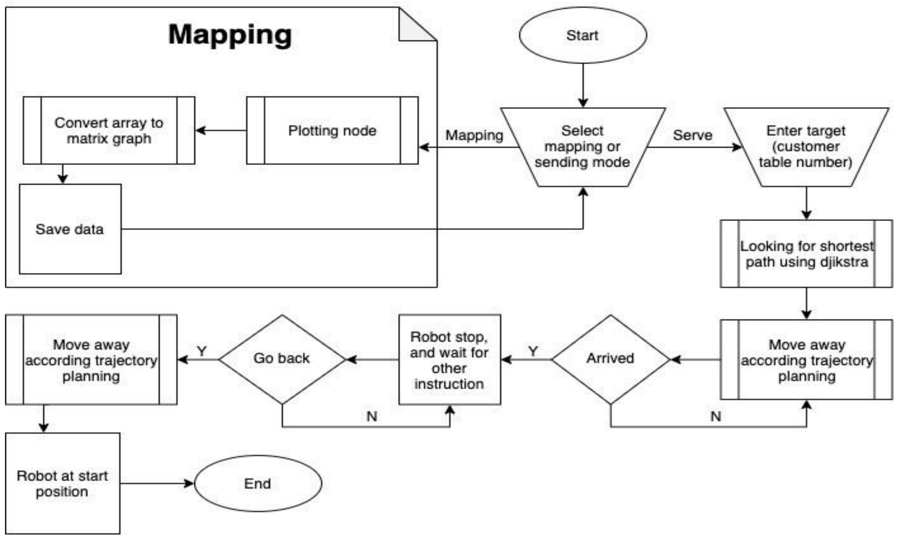
Mapping
Start
Select mapping or sending mode
Mapping
Serve
Enter target (customer table number)
Plotting node
Convert array to matrix graph
Save data
Looking for shortest path using djikstra
Move away according trajectory planning
Arrived
Y
N
Robot stop, and wait for other instruction
Go back
Y
N
Move away according trajectory planning
Robot at start position
End

Figure 2. Flowchart System of Mapping Process

If the work area is not a new area, then the robot works as in Figure 2. The user can press the 'serve' button and the table number of the intended customer. Then the robot will move towards the target via the shortest path that has been mapped. And if the robot has arrived at the target, the customer can take the order and send the robot back by pressing the 'back' button, and the robot will automatically return to the starting place (kitchen) to carry out the next task.

3. Odometry

Odometry is used to estimate the coordinates of the position relative to the initial position [13], [14]. In the wheeled robot odometry system, the sensor used is a rotary encoder which calculates the number of wheel rotations. There are three main parameters in calculating the coordinates of the robot position [15], namely the diameter of the free wheel wheel (DW), the number of encoder resolutions (〖resolution〗 _enc), and the number of pulses generated by the encoder sensor (pulse). To find out the change in coordinates in millimeters (mm), it is necessary to know in advance the circumference of the free wheel (KW) wheel has been presented in equation (1), after that to find out the position of X (xTempuh), and Y (yTempuh) coordinates can be calculated using equations (2) and (3).

$$KW = DW \times \pi \quad (1)$$

$$xTempuh = \frac{pulse_x}{resolusi_{enc}} KW \quad (2)$$

$$yTempuh = \frac{pulse_y}{resolusi_{enc}} KW \quad (3)$$

For the bearing target (β) it can be calculated by equation (4), while to find out the value of the distance (distance) from the current point to the target point, the data is calculated by equation (5).

$$\beta = \tan^{-1} \frac{yTujuan - yTempuh}{xTujuan - xTempuh} \quad (4)$$

$$jarak = \sqrt{(xTujuan - xTempuh)^2 - (yTujuan - yTempuh)^2} \quad (5)$$

4. Dijkstra

Based on the mapping data that has been done, to find the shortest path so that orders arrive at the customer quickly, this research requires an optimization method. In this study, the search for the shortest path was using Dijkstra's algorithm [16]. Dijkstra's algorithm is used to determine the shortest path [17], [18] which will be used as the planning trajectory by the robot to run. The Dijkstra process requires some data, including (1) Path graph (2) Weights between nodes or vertices. Graph data is obtained from the results of the mapping of the positions of all nodes that have been done. Dijkstra's algorithm performs calculations on all possibilities to find the smallest weight from one node to another. Figure 3 (a) is an example of Dijkstra's implementation using a path graph design. Table 1 is the process of calculating Dijkstra's algorithm in the form of data planning trajectory.

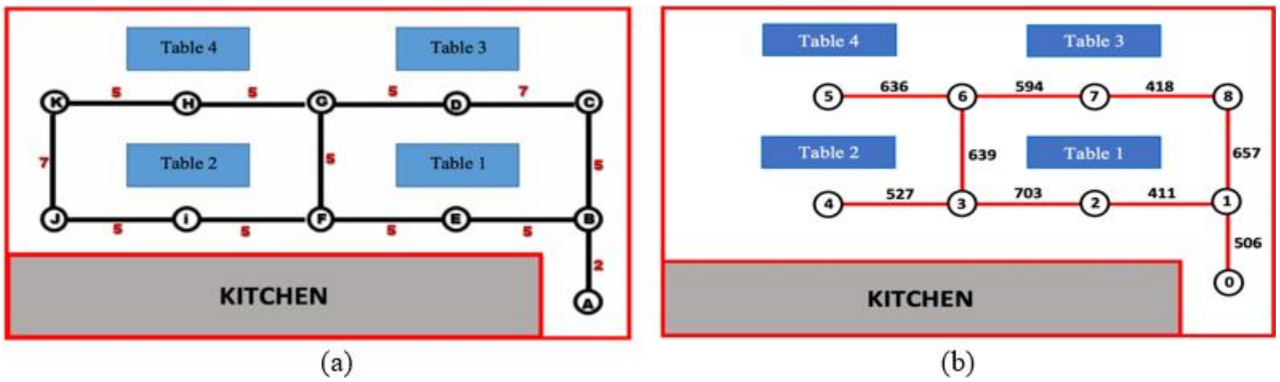

Figure 3. (a) Plan and Path Graph; (b) Path Graph of Dijkstra's Calculation Result

Table 1. Dijkstra's calculations for the shortest path

No	Unvisite	Visited	Current	A	B	C	D	E	F	G	H	I	J	K
0	ABCDEFGHIJK	~	~	$(0,\sim)_n$	$(\infty,\sim)_n$	$(\infty,\sim)_n$	$(\infty,\sim)_n$	$(\infty,\sim)_n$	$(\infty,\sim)_n$	$(\infty,\sim)_n$	$(\infty,\sim)_n$	$(\infty,\sim)_n$	$(\infty,\sim)_n$	$(\infty,\sim)_n$
1	BCDEFGHIJK	A	A		$(2,A)_1$	$(\infty,A)_1$	$(\infty,A)_1$	$(\infty,A)_1$	$(\infty,A)_1$	$(\infty,A)_1$	$(\infty,A)_1$	$(\infty,A)_1$	$(\infty,A)_1$	$(\infty,A)_1$
2	CDEFGHIJK	AB	B			$(7,B)_2$	$(\infty,B)_2$	$(7,B)_2$	$(\infty,B)_2$	$(\infty,B)_2$	$(\infty,B)_2$	$(\infty,B)_2$	$(\infty,B)_2$	$(\infty,B)_2$
3	DEFGHIJK	ABC	C				$(14,C)_3$	$(7,B)_2$	$(\infty,C)_3$	$(\infty,C)_3$	$(\infty,C)_3$	$(\infty,C)_3$	$(\infty,C)_3$	$(\infty,C)_3$
4	DFGHIJK	ABCE	E				$(14,C)_3$		$(12,E)_4$	$(\infty,E)_4$	$(\infty,E)_4$	$(\infty,E)_4$	$(\infty,E)_4$	$(\infty,E)_4$
5	DGHIJK	ABCEF	F				$(14,C)_3$			$(17,F)_5$	$(\infty,F)_5$	$(17,F)_5$	$(\infty,F)_5$	$(\infty,F)_5$
6	GHIJK	ABCEFD	D							$(17,F)_5$	$(\infty,D)_6$	$(17,F)_5$	$(\infty,D)_6$	$(\infty,D)_6$
7	HIJK	ABCEFDG	G								$(22,G)_7$	$(17,F)_5$	$(\infty,G)_7$	$(\infty,G)_7$
8	Hxz JK	ABCEFDGI	I								$(22,G)_7$		$(22,I)_H$	$(\infty,I)_H$
9	JK	ABCEFDGIH	H										$(22,I)_H$	$(27,H)_9$
10	K	ABCEFDGIHJ	J											$(27,H)_9$
11														

The results of calculating the distance from the selected node to its neighboring nodes are stored in a variable, then the results are used as weights for calculations in Dijkstra. If the selected node has no neighbors, the value is equal to 0. The calculation will continue to be made to all nodes. The mapping results are then converted into a path graph as shown in Table 2. Figure 3 (b) is the result of the path graph that has been calculated using Dijkstra. Furthermore, this path graph will be used by the robot in delivering food to the customer's table.

Table 2. Graph for the Path

	0	1	2	3	4	5	6	7	8
1	0	506	0	0	0	0	0	0	0
2	506	0	411	0	0	0	0	0	657
3	0	411	0	703	0	0	0	0	0
4	0	0	703	0	527	0	639	0	0
5	0	0	0	527	0	0	0	0	0
6	0	0	0	0	0	0	636	0	0
7	0	0	0	639	0	636	0	594	0
8	0	0	0	0	0	0	594	0	418

5. Experimental Result and Discussion

ISSN: 04532198
Volume 62, Issue 11, December, 2020

The first thing that must be ensured before the robot delivers the order to the customer table is the robot's ability to walk straight. This test must be done because the robot moves without a track marker. In addition, it aims to determine the error heading. Testing is done by giving orders to the robot to walk straight forward 150cm. In Figure 4 (a) the robot moves using the IMU sensor and Figure 4 (b) the robot moves without the IMU sensor.

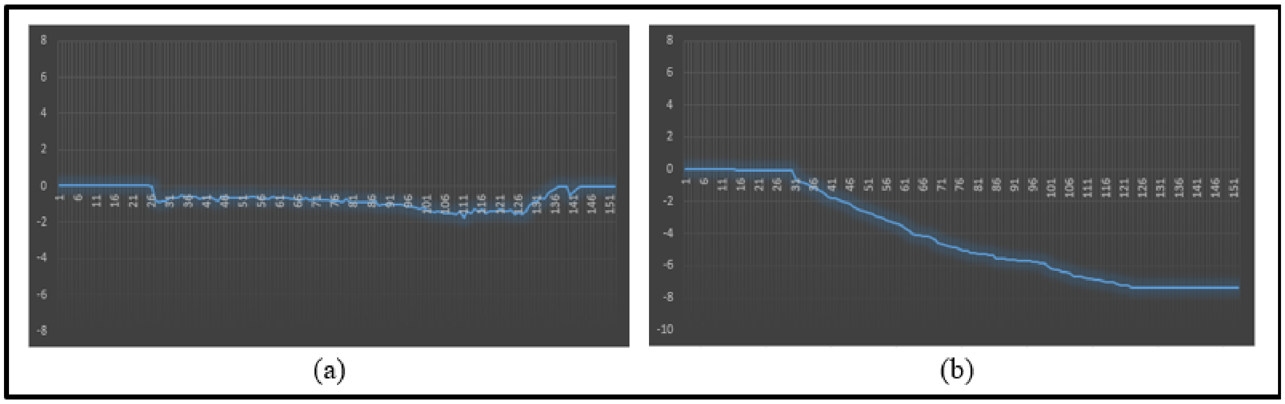

Figure 4. (a) Using IMU Sensor; (b) Without IMU Sensor

Based on the mapped path, the waiter robot trial has been carried out in the laboratory with the position adjusted as in a restaurant environment as in Figure 5. From several experiments the robot has succeeded in delivering food to the customer's table using the shortest path. For example, orders must be delivered to table 3, then the path chosen by the robot is 0-1-8-7 with a route length of 1581. No food and drinks are delivered even though there is a vibration from the robot mechanic.

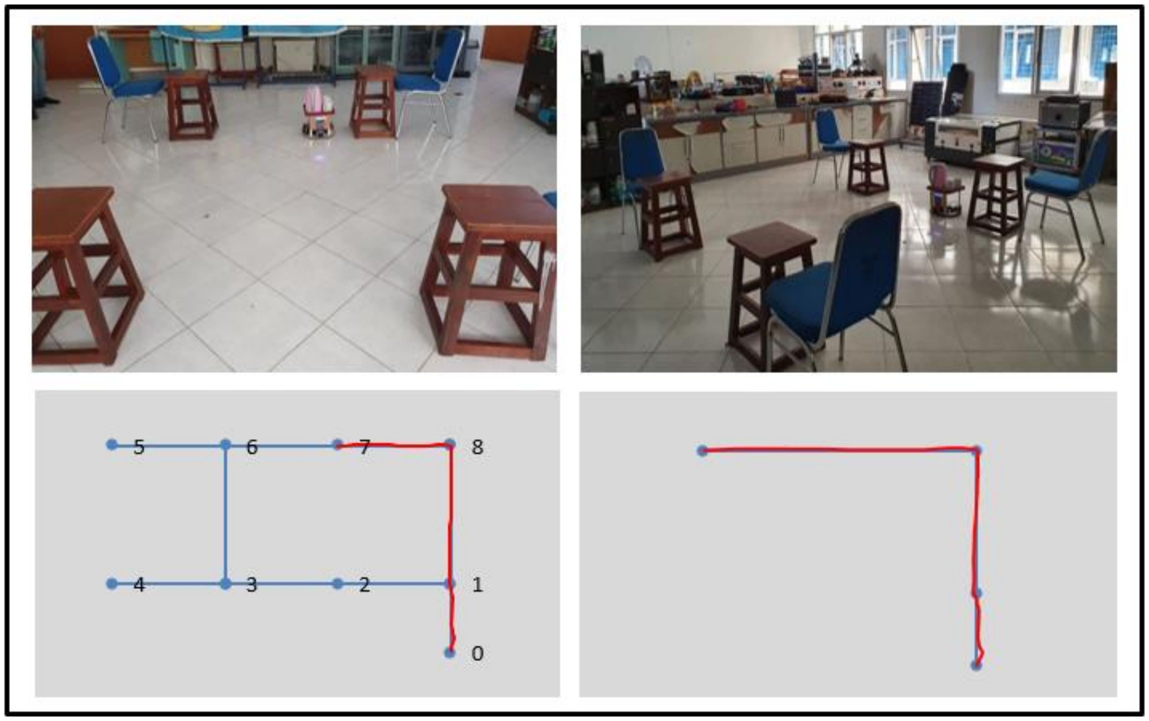

Figure 5. Robot Trial

6. Conclusion

Autonomous Waiter Robot that uses omni-directional wheels to produce smooth movement. The application of the odemetry method and the djikstra algorithm is running well. The robot moves to the destination table with the shortest route, the fastest. The IMU sensor is very helpful in determining the robot heading parameters, so that the robot remains in the correct position on the destination table and does not rotate. This automatic food delivery robot will continue to run in the track route position according to the initial mapping results, even though the robot is tried to be kept away from the mapping route.

7. References

[1] Z. H. Khan, "Waiter Robot – Solution to Restaurant Automation," no. November 2015, 2016.

[2] T. L. M. Santos, "ScienceDirect Trajectory tracking of Omni-directional Trajectory tracking of Trajectory tracking of Omni-directional Mobile Robots via Predictive Control Plus Mobile Robots via Predictive Control Mobile Robots via Smith Predictive Control plus plus a Filtered Predictor a Filtered Smith Predictor a Filtered Smith," vol. 1, pp. 10250–10255, 2017.

[3] D. J. Nibin, J. V Ajesh, and N. V Ajin, "Smart food serving robot in restaurant 1," pp. 151–154, 2019.

[4] A. Cheong, M. W. S. Lau, E. Foo, J. Hedley, and J. W. Bo, "Development of a Robotic Waiter System Development of a Robotic Waiter System," IFAC-PapersOnLine, vol. 49, no. 21, pp. 681–686, 2016.

[5] N. T. Newaz, "IoT Waiter Bot: A Low Cost IoT based Multi Functioned Robot for Restaurants," pp. 1174–1178, 2020.

[6] "Design and Implementation of a Robotic Technique Based Waiter," no. December, pp. 7–9, 2017.

[7] R. Saple, "Robotic Waiter," vol. 1, no. 11, pp. 251–254, 2015.

[8] N. Malik, A. Singh, and N. Rani, "Serving Robot: New Generation Electronic Waiter," vol. 6, no. 4, pp. 3763–3766, 2016.

[9] Z. Abbas, S. Ahmed, S. Abbas, and A. Mazhar, "Automatic Cafe Management System Using Waiter Robot," pp. 211–214, 2019.

[10] B. Adamov, "INFLUENCE OF MECANUM WHEELS CONSTRUCTION ON ACCURACY OF THE OMNIDIRECTIONAL PLATFORM NAVIGATION (ON EXANPLE OF KUKA YOUBOT ROBOT) *," no. May, 2018.

[11] S. Guo, Q. Diao, and F. Xi, "Vision Based Navigation for Omni-directional Mobile Industrial Robot," Procedia - Procedia Comput. Sci., vol. 105, no. December 2016, pp. 20–26, 2017.

[12] Susan Park, Y. Ryoo and D. Im, "Fuzzy steering control of three-wheels based omnidirectional mobile robot," 2016 International Conference on Fuzzy Theory and Its Applications (iFuzzy), Taichung, 2016

[13] A. Ligocki, A. Jelínek, Fusing the RGBD SLAM with Wheel Odometry, IFAC Papers On Line, Volume 52, Issue 27, 2019.

[14] Y. Tanaka, A. Semmyo, Y. Nishida, S. Yasukawa, J. Ahn and K. Ishii, "Evaluation of underwater vehicle's self-localization based on visual odometry or sensor odometry," 2019 14th Conference on Industrial

and Information Systems (ICIIS), Kandy, Sri Lanka, 2019

[15] T. Bräunl. Embedded Robotics Mobile Robot Design and Applications with Embedded Systems. 3nd Edition (english version), Springer-Verlag, Berlin, Heidelberg, 2008

[16] J. Russell, R. Cohn. Dijkstra's Algorithm. Book on Demand. 2012

[17] Z. Nie and H. Zhao, "Research on Robot Path Planning Based on Dijkstra and Ant Colony Optimization," 2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Shanghai, China, 2019

[18] O. A. Gbadamosi and D. R. Aremu, "Design of a Modified Dijkstra's Algorithm for finding alternate routes for shortest-path problems with huge costs.," 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), Ayobo, Ipaja, Lagos, Nigeria, 2020