Design and Construction a Dynamic Store System using the Bezier Curve Algorithms

Abdulmuttalib T. Rashid
Electrical Engineering Dep.
University of Basrah
Basrah, Iraq

Fatima R. Ali
Electrical Engineering Dep.
University of Basrah
Basrah, Iraq

Osama T. Rashid
Computer Engineering Dep.
Iraq University Collage
Basrah, Iraq

ABSTRACT

The objective of this paper is to design a dynamic store system consist of several boxes that arranged in several columns according to the N-ary tree algorithm. Each box is designed to store and return small object using the differential drive mobile robot according to a secret code that entered from a user interface unit. In a dynamic store system the robot has changeable trajectories according to the selected store box. The chosen trajectory is highlight by a group of LEDs connected a long this path. The mobile robot is equipped by photocells to work as line follower sensors. The Bezier curve algorithms are used in this method to reduce the rabble occurs when the mobile robot moves on the line follower system. The dynamic method is simulated and implemented practically to compare with the best static method for choosing the one with the best performance in the time of arrival and the average length of the paths.

General Terms

Differential drive mobile robot, Object store system.

Keywords

Dynamic store system, N-ary tree algorithm, Bezier curve algorithm, line follower robot.

1. INTRODUCTION

An automated stored/retrieved system (AS/RS) is a necessary system in automation techniques because of it is ability to provide the reliability and the scalability in store and retrieve the objects [1]. The efficient operation of AS/RS requires the planning of [2]:

- Physical storage specification: width, length, height of storage structure.
- Operating characteristics of AS/RS system: velocity, Acceleration rate and the number of machines.
- Control strategy.

The control of the robots motion can be classified into three motion tasks as in the following points [3]:

- Point-to-point movement: In this task an initial location and another target location are set, so the robot movement starts from this initial position and continues to moves until reaches to the target.
- Path following: In this task, the line follower robot is following the path that drawn on the ground.
- Trajectory tracking: In this task also the mobile robot must arrive and follow a path or a trajectory in a Cartesian space.

Line follower robots can be used in motion trajectory to design the object storage system [4]. The line follower robot can be follow a path with a white color drawn on a black surface, or follow a path with a black color drawn on a white surface [5].

Most of the old version of the AS/RSs are designed to control problems in static environments which are no suitable to work with the rapidly products' demands, short time internet orders, and high service level requirements. This leads the researchers to move towards developing prototypes, and algorithms that include the dynamic aspects in the AS/RSs. These aspects represent by self-adaptive storage assignment methods, dynamic dwell-point rules and also, focus more on robustness of the design. The object storage systems can be designed by using either static or dynamic environment. In the static environment, only the robot has the ability of movement and the other objects are static where in the dynamic environment, as well as the robot, there are some other objects can also be considered as dynamic (such as moving machines, moving paths) [6]. In 2014, J. Gagliardi, et.al [7], design and improve an automated storage and retrieved system. The operation way of this system wills effects on its efficiency. Also this design discussed that the performance of the system depend on the sequence of the request (the stored and retrieved operations perform According to a specific order). In the designed system the problem of sequencing and storage location solved by using a "sequencing mathematical model" (SMM). The drawback of the SMM it is not considered the best solution for all the designed storage systems. In an Amazon Company the Kiva robot are used to move the shelving units from where its stored to the ordered location, and then this robot restore this units to its storage location [8, 9]. In year 2013, Amazon Prime Air system is introduced by the same company, this system defined as an automated supply system. It used the miniature unmanned air vehicles (MUAVs) to supply suites to the people [10]. The motion trajectory of the pick-and-place robot is programmed to quick and repetitive picking of objects with the same weights and sizes [11, 12].

This paper deals with the enhancement the length of the paths and the time of arrival in the object store system by representing the moving trajectories by the Bezier curve algorithms. There are some difficulties in using the static environment to represent the Bezier curve trajectories because this leads to interface these trajectories. The solving of this problem is achieved by replace the static environment by the dynamic environment. In dynamic environment the N-ary tree paths are changed from black lines to the lighting lines. Only the chosen path for movement is lighted and the other paths are turn off. An electronic device must be design and construct to implement the dynamic environment.

2. THE DYNAMIC STORE SYSTEM

This section deals with the design of the dynamic store system to enhancement the length of the paths and the time of arrival in the object store system by representing the moving trajectories by the Bezier curve algorithm. There are some difficulties in using the static environment to represent the Bezier curve trajectories because this leads to interface these trajectories. In dynamic environment the paths are changed from black lines to the lighting lines. Only the chosen path for movement is lighted and the other paths are turn off. The dynamic environment means that there is a change happening in the environment at each store or restore operation. The original shape of the environment before applying the dynamic algorithm is shown in Fig 1. In this system the N-ary tree algorithm is used to construct the environment with N=3, where N represents the maximum number of children at each branch.

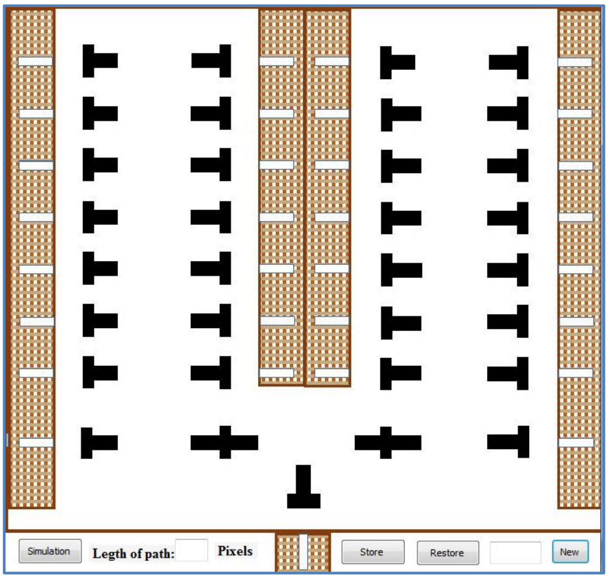


Fig 1 The initial shape of the dynamic environment.

From Fig 1 we found that the storage environment is divided into two symmetrical groups of boxes and each one has 15 boxes. Each storage box has a unique code number consist of five binary digits that used to access to this box. This unique code represents by three parts: Dir1, dir2 and cross. The first bit (LSB) represents the Dir1, the second bit represents Dir2 and the last three bits (MSB) represent cross. All these three parts are used by the robot to access to the desired box. When Dir2=0 the robot rotate to the right group of boxes else it rotates to the left and when Dir1= 0 the robot rotate to the right sub group of each group else it rotates to the left. The number of cross in the environment is selected by the Cross bits. The access codes and states of all storage boxes are stored in random access file to control the process of storing and retrieving the objects from each box. Each box has a record in the file for it is information. The information consists of unique access code (Dir1, Dir2, and Cross) and one binary bit represents the states of this box (tst). If tst = 0 then the box is empty else it full. The process of storing any object accomplishes by opening the random access file and searching sequentially for box has tst = 0 (empty box), and when it found then the tst code changed to tst = 1. The reverse operation is doing when we want to retrieves any object from any box.

In this system the differential drive mobile robot (DDR) is used [13, 14]. This robot is one of the mostly used classes of wheeled mobile robots with two active wheels that are driven by different motors for controlling the robot. To turn the robot left or right, wheels are rotated at “different” speeds or in “different” directions. The DDR used in this method contains two line follower sensors (right and left photocell sensors). Fig 2 shows the complete flow chart for store and retrieve an object using the dynamic object store system.

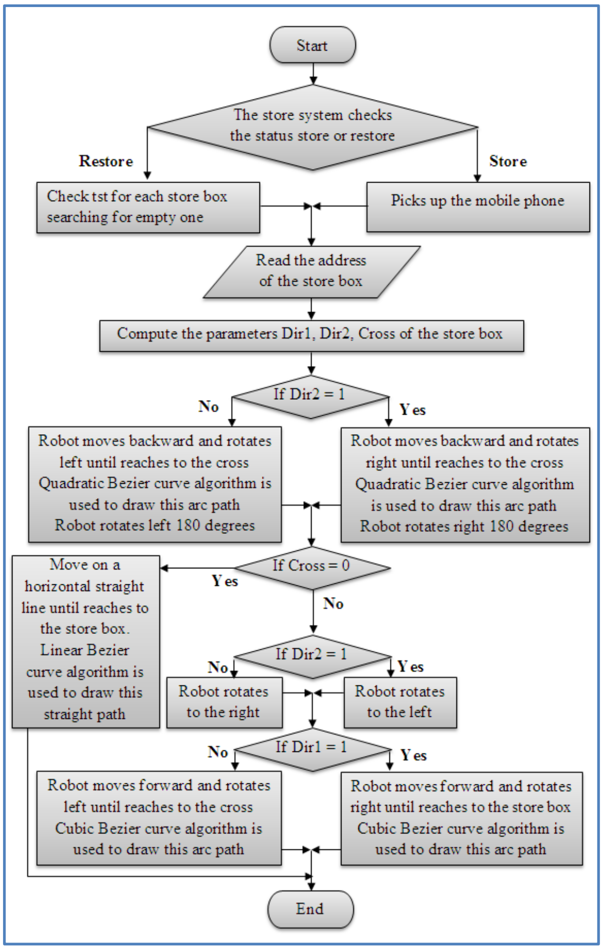

Start
The store system checks the status store or restore
Restore
Store
Check tst for each store box searching for empty one
Picks up the mobile phone
Read the address of the store box
Compute the parameters Dir1, Dir2, Cross of the store box
If Dir2 = 1
No
Yes
Robot moves backward and rotates left until reaches to the cross Quadratic Bezier curve algorithm is used to draw this arc path Robot rotates left 180 degrees
Robot moves backward and rotates right until reaches to the cross Quadratic Bezier curve algorithm is used to draw this arc path Robot rotates right 180 degrees
If Cross = 0
Yes
No
Move on a horizontal straight line until reaches to the store box. Linear Bezier curve algorithm is used to draw this straight path
If Dir2 = 1
No
Yes
Robot rotates to the right
Robot rotates to the left
If Dir1 = 1
No
Yes
Robot moves forward and rotates left until reaches to the cross Cubic Bezier curve algorithm is used to draw this arc path
Robot moves forward and rotates right until reaches to the store box Cubic Bezier curve algorithm is used to draw this arc path
End

Fig 2 The flowchart to the dynamic store system

3. STORE SYSTEM CONSTRUCTION

The section deals with construction a dynamic store system using Bezier curve algorithms. A Bezier curve is defined by a set of control points P_0 through P_n, where n is called its order (A linear Bezier curve is simply a straight line between two points, a quadratic Bezier curve is the path traced by three given points and The cubic Bezier curve has four points P_0, P_1, P_2 and P_3). The Bezier curve algorithm is used to make the trajectories smooth and continuous [15]. Two main parts include in this system: The line follower robot and the structure and electronic circuits of the dynamic environment.

3.1 The line follower robot

The robot used in this system is a differential drive mobile robot as shown in Fig 3. There is one difference between this robot and the standard line follower robot which represents by the line follower sensors. In the standard one, two pairs of Infra-red sensors are used to detect the black paths drawn on the static environment where in this robot the Infra-red sensors are replaced by a pair of photocell resistors to detect the lighting paths which changed dynamically according to the select store box.

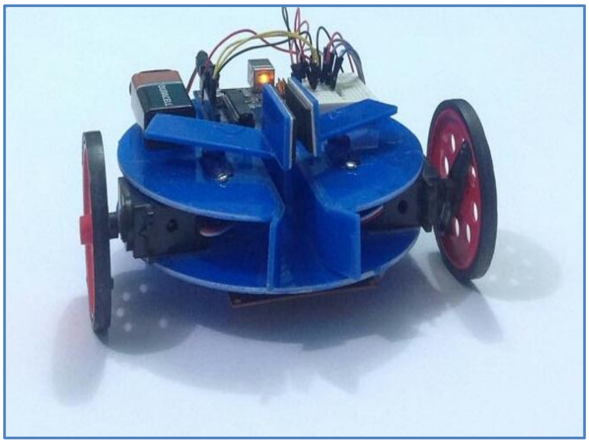


Fig 3: The differential drive mobile robot.

The photocells are small, low power, inexpensive and easy to use sensors which used to detect the light. They are used in street lamps that turn themselves on at night. Sometimes they knew as photo resistors. They look like a small (0.5 to 2 inch) disk with two terminals as shown in Fig 4. In operation, a photocell looks like a light-sensitive resistor with a high resistance up to 10 MΩ when dark and a low resistance when in the light.

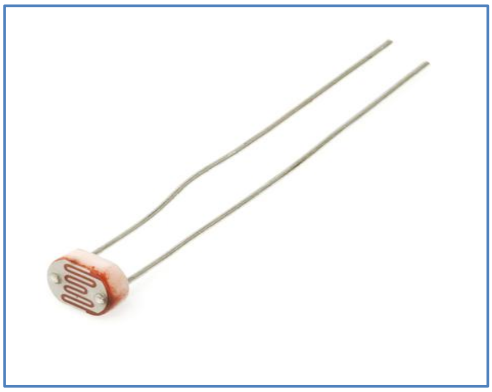


Fig 4: The photocell sensor.

The photocell sensors board and the schematic diagram of it is connection to the microcontroller of the robot is shown in Fig 5 and 6 respectively.

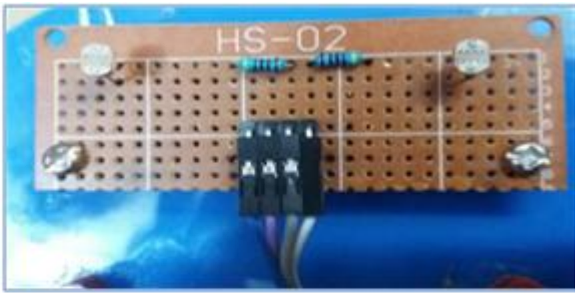


Fig 5: The photocell sensor board.

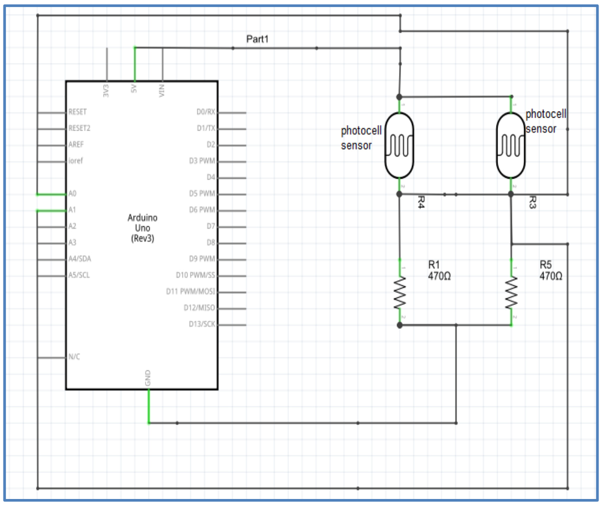
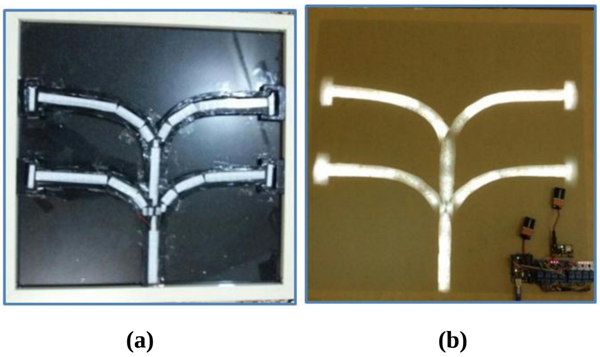


Fig 6: The schematic diagram to the photocell sensor board connection.

This robot uses the program which described by the same flow chart in Fig 2 for store and retrieves an object using the dynamic object store system.

3.2 Construction the dynamic environment

The dynamic environment means that the tree of paths in object store system is not fixed and can be change according to the destination store box. The reason for making this environment changeable because the trajectories to all store boxes are drawn by the Bezier curve algorithms and this make them intersect with each other when drawn as fixed paths. The solving of this problem is by making these trajectories as paths of light and only the destination path is lighted. According to this requirement the dynamic environment of the store system has two layers. The lower one (Fig 7 (a)) has four paths of light isolated from each other and drawn by using the Bezier curve algorithm and the upper layer (Fig 7 (b)) is represent by 80 * 80 cm plastic board which capable to pass the light of the lower layer.



(a) **(b)**

Fig 7: The dynamic environment of the object store system. (a) Lower layer. (b) Upper layer.

Fig 8 and 9 show the device that used for controlling the lights of the dynamic environment paths and it is schematic diagram respectively. This device has one Arduino Uno as a microcontroller, two kits of 4-relay module, 3.3 V / 5 V power adapters for driving the relay module and two 9-Volt batteries. The schematic diagram shows that the control device drives eight lines of lights. These lines are divided into four groups to produce four smooth trajectories of light as shown in Fig 10. Table 1 shows the four paths with their group of lines.

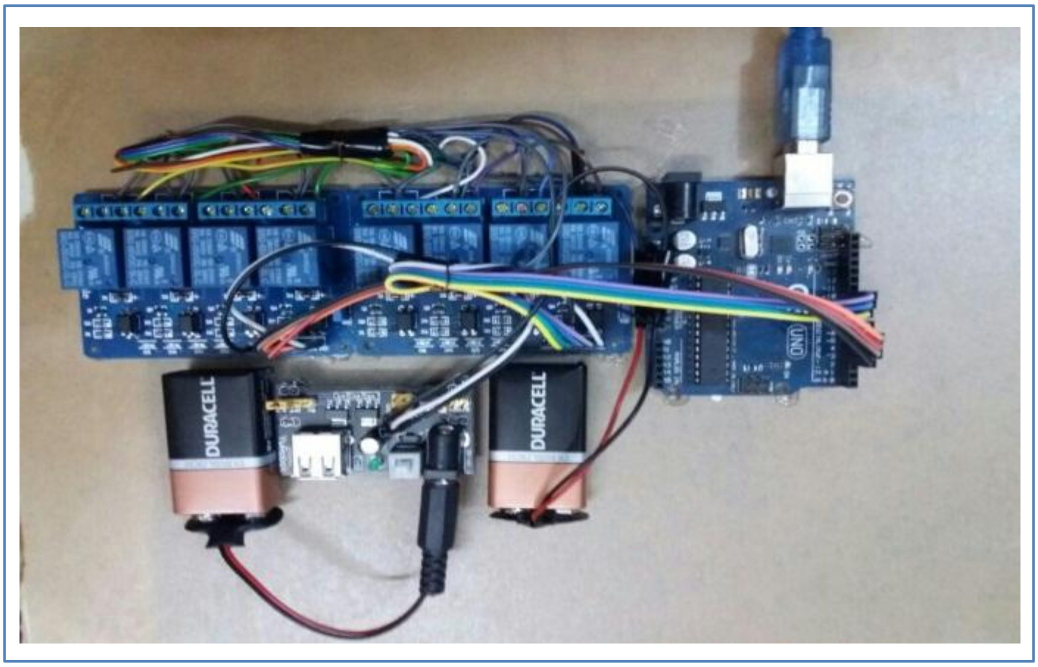


Fig 8: The control device to the lights of the dynamic environment paths.

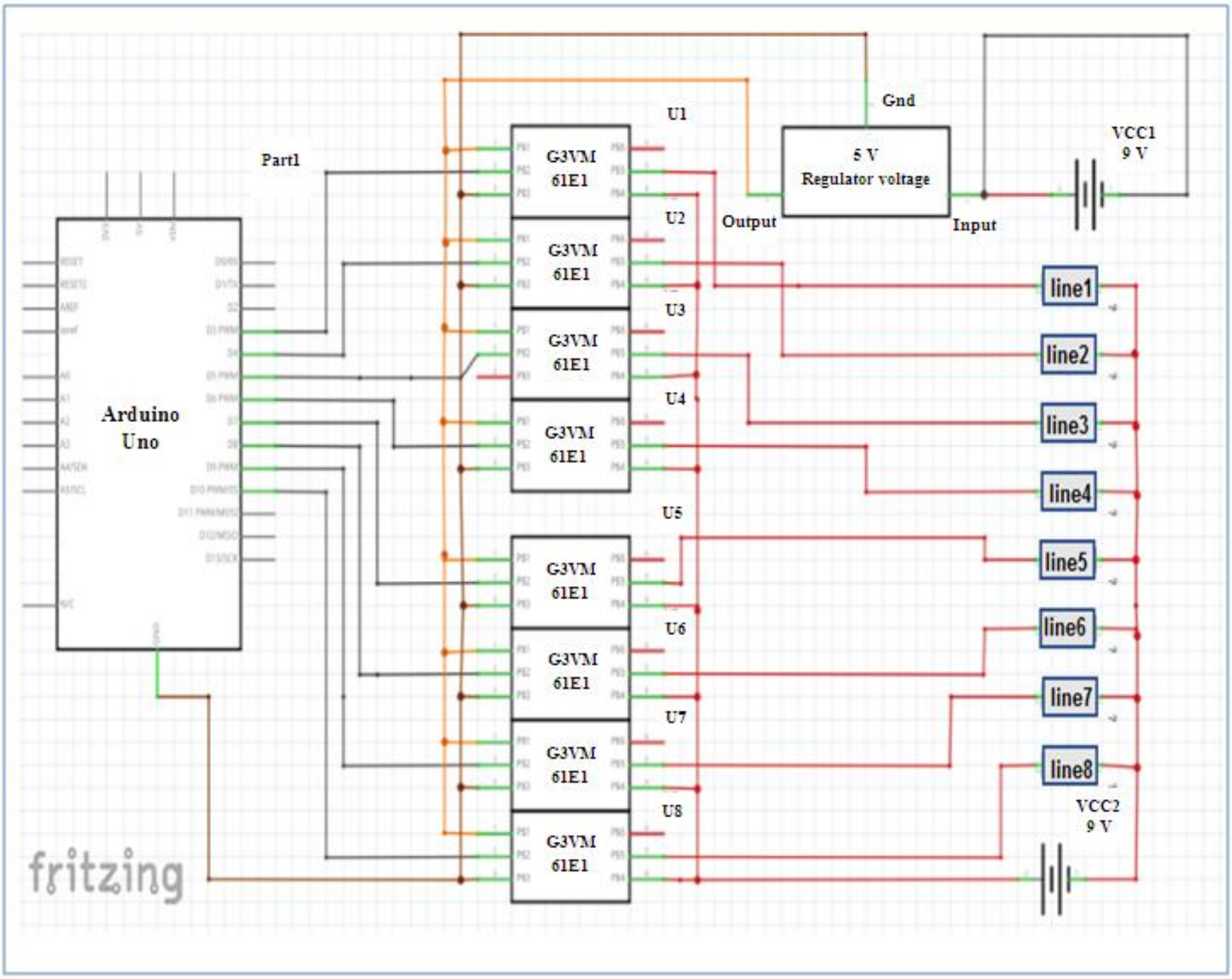


Fig 9: schematic diagram to the device used to control the paths of light.

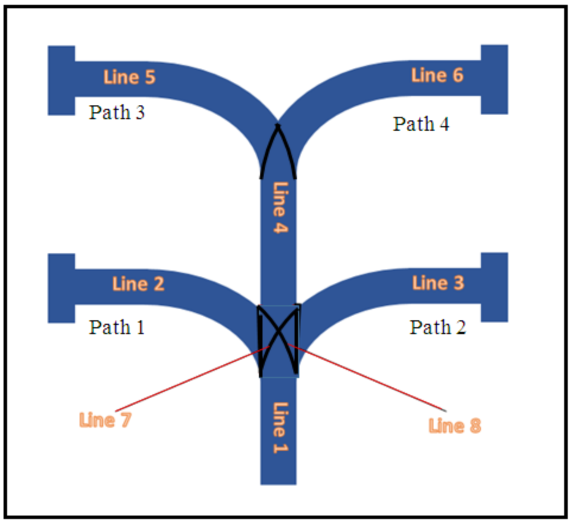


Fig 10 Schematic shows the four paths of lights with their group of lines.

Table 1 The four paths of lights with their group of lines.

	Paths of lights	Group of lines
1	Path1	Line1, Line2, Line7
2	Path2	Line1, Line3, Line8
3	Path3	Line1, Line4, Line5, Line7, Line8
4	Path4	Line1, Line4, Line6, Line7, Line8

Each line of these eight lines has several LEDs connected in parallel and requires high current to lighting. For this purpose, two 4-Relay modules are used to drive the LEDs of these lines (One relay to each line). The relay is an electromagnetic switch used to drive high current and voltage devices like motors and lights using low power signals. Fig 11 shows the 4-Relay module which requires four control signals from microcontroller and has four normally close - normally open connectors one for each relay. The supply voltage for these modules is the 3.3 / 5 Volts power adapters (Fig 12) which provide both 3.3 and 5 volts and requires 9-volt battery to work. Another 9-volt battery is used for lighting the lines of LEDs and connected through the 4-Relay modules as shown in Fig 9.

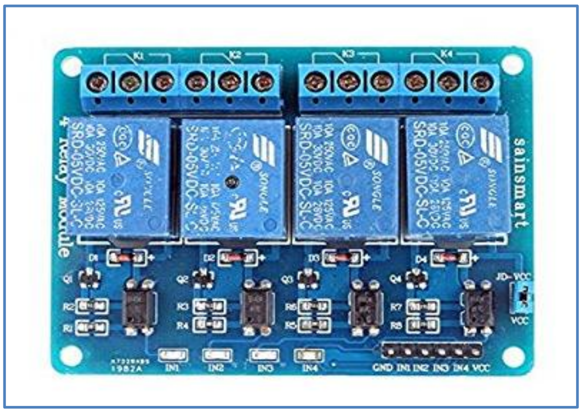


Fig 11: 4-Relay modules.

Fig 12 3.3 / 5 volts power adapter.

The final shape of the dynamic environment object store system is shown in Fig 13.

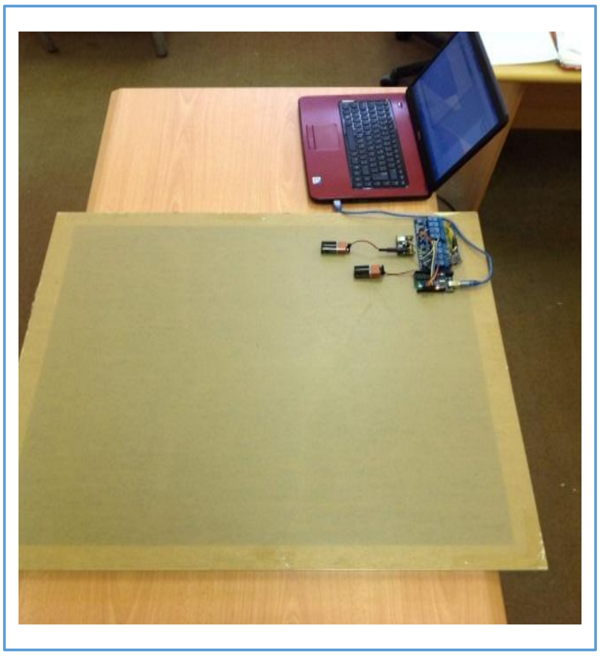


Fig 13: The dynamic environment objects store system.

The device of the dynamic environment object store system is a computer based device because it receives the orders from the computer through the serial port. The Arduino kit which used as a microcontroller by this device is connected through the USB port of the computer and controlled by a program written by the visual basic programming language version 6. Fig 14 shows the graphical user interface (GUI) of this program. This GUI has four paths similar to the actual paths on the board of the dynamic environment and four control Buttons titled by the name of paths. Each path on the GUI is drawn by some shapes object in visual basic language. Each control button in GUI has two tasks: The first one is to make the visibility property of the selected path (the shapes object of this path) equal to True and the second task is to gives an order through the serial port to the device of the dynamic environment to lighted the selected path. Fig 15 and 16 show the effect of clicking the Button path 2 on the GUI of the program and the board of the dynamic environment.

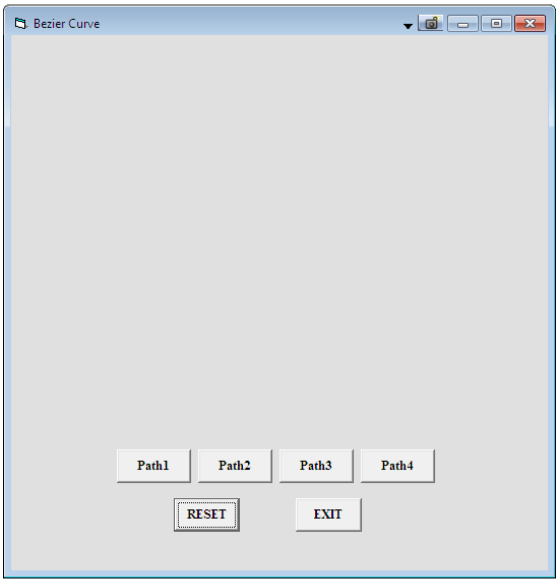


Fig 14 The GUI of the dynamic environment controlled program.

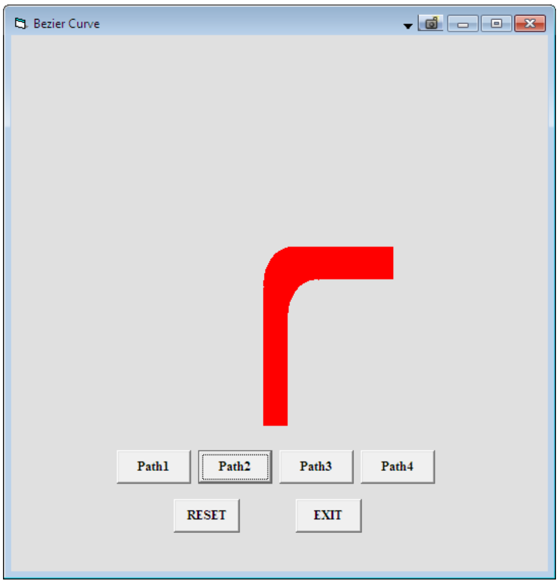


Fig 15 The path2 selecting on the GUI of the dynamic environment.

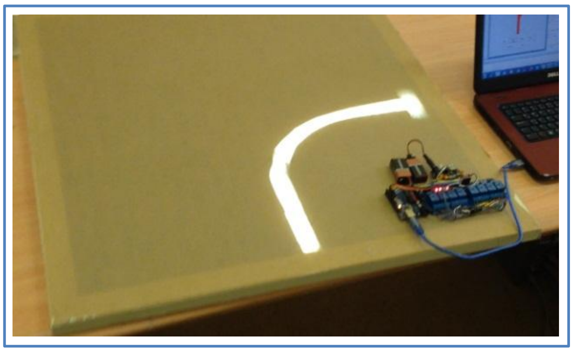


Fig 16: The effect of clicking path2 Button in the GUI on the board of the dynamic environment.

Fig 17 shows the flowchart for the visual basic program that written to control the device of the dynamic environment board.

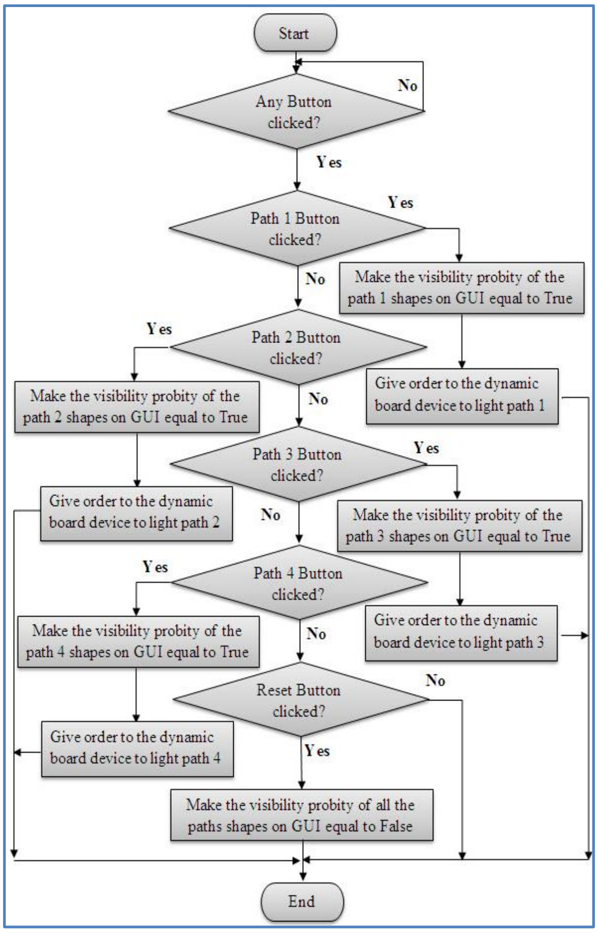

Start
Any Button clicked?
No
Yes
Path 1 Button clicked?
Yes
No
Make the visibility probity of the path 1 shapes on GUI equal to True
Give order to the dynamic board device to light path 1
Path 2 Button clicked?
Yes
No
Make the visibility probity of the path 2 shapes on GUI equal to True
Give order to the dynamic board device to light path 2
Path 3 Button clicked?
Yes
No
Make the visibility probity of the path 3 shapes on GUI equal to True
Give order to the dynamic board device to light path 3
Path 4 Button clicked?
Yes
No
Make the visibility probity of the path 4 shapes on GUI equal to True
Give order to the dynamic board device to light path 4
Reset Button clicked?
No
Yes
Make the visibility probity of all the paths shapes on GUI equal to False
End

Fig 17 The flowchart of the program that used to control the device of the dynamic environment board.

4. EXPERIMENTAL RESULTS

In this paper, the dynamic object store system is constructed using Bezier curve algorithms. Differential drive mobile robot was assumed in our experiments. These experiments are used to measure the performance of the suggested method with respect to the linear object store system [16]. The linear paths method works with static environment and produces less length of the path and less time of arrival with respect to other suggested methods [17]. The experimental results are tested on environment has four dynamic paths. Fig 18 shows the steps for the movement of the Differential drive mobile robot on the third path of the dynamic environment board. This experiment is repeated for all the paths on this board and compared with the experiment implemented on the static object store system. These experimental are used to measure the performance for both the length of the path and the time of arrival. The time of arrival for the two methods is the time (in second) that the robot needs it to access to the box and then back to its center. The length of the path is the number of pixels that the robot needs to reach to the storage box and return to the center [16, 17]. Fig 19 shows the relationship between the number of storage boxes and the time of arrival (Sec) for both methods. The second comparison shows the relationship between the number of storage boxes and the average length of paths (Cm) as shown in Fig 20. From the first comparison we found that the dynamic environment object store system has the shortest time of arrival. The second comparison shows that the dynamic environment object store system has the shortest path.

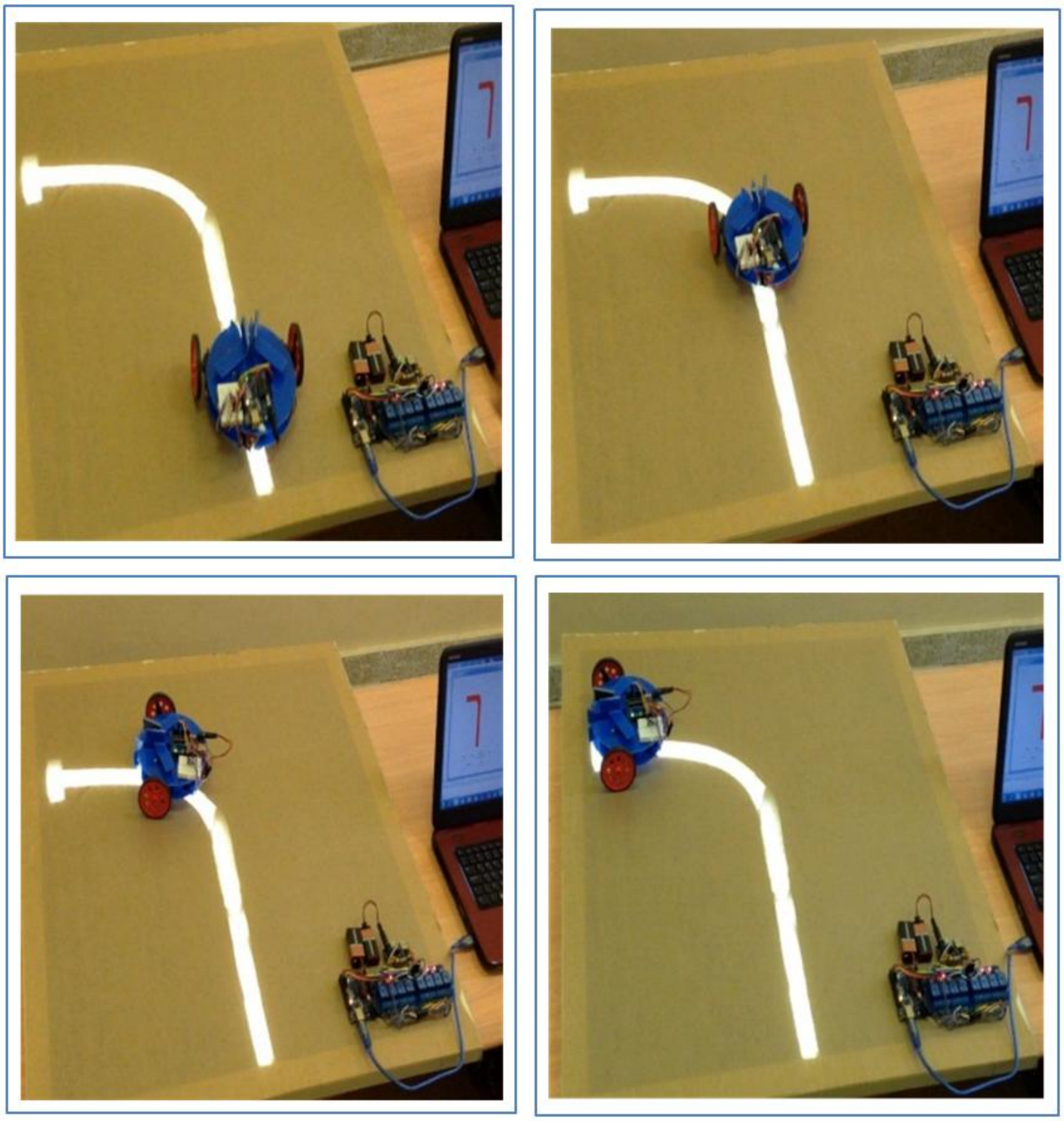


Fig 18 The illustration of the robot movement on the third path of the dynamic environment board.

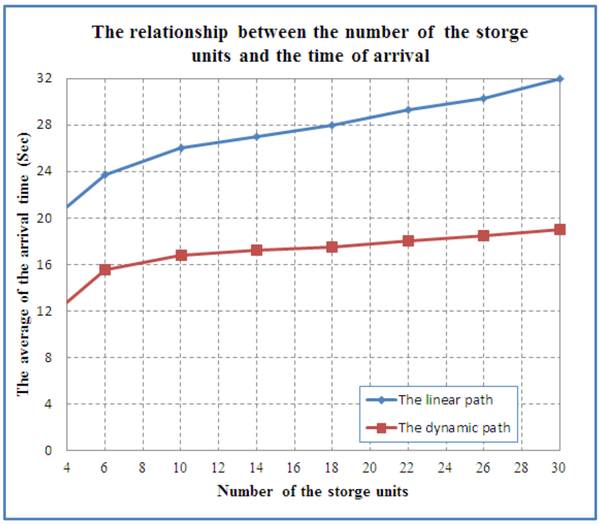


Fig 19 Experimental results show the relationship between the number of the storage units and the time of arrival.

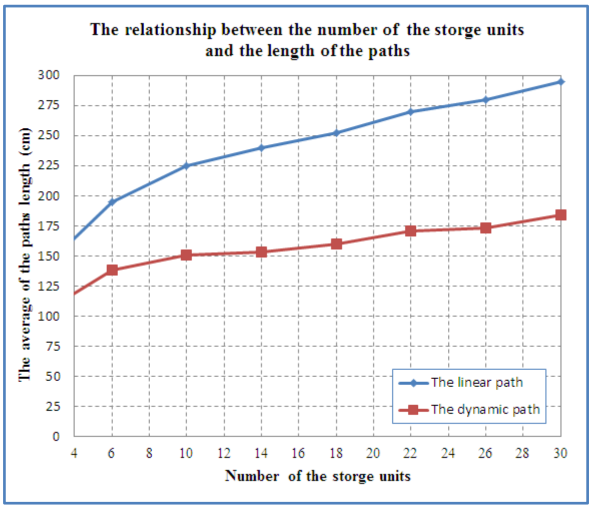


Fig 20 Experimental show the relationship between the number of the storage units and the length of the paths.

5. CONCLUSION

In this paper a dynamic environment object store system is suggested to work as store and retrieves system for small objects. It is designed and constructed in hardware. This system is compared with static environment object store system [16, 17]. From the experimental results we found the dynamic environment system has the best performance in both the average length of paths and in the time of arrival. Also the dynamic environment paths store system produces only one path to the robot. This is done because of using the Bezier curve algorithm to enhance the trajectory of movement. The enhancement occurs because the length of the paths are reduced and also, the cross points are removed in the dynamic environment. This suggested system can improve in future work by using multi-layer of the store boxes to increase the capacity of the system and also using more than one robot in the system reduce the time for storing and restoring operations.

6. REFERENCES

[1] L. Wu, Q. Zhuge, E. H. Sha, X. Chen, and L. Cheng, "An Efficient Data Distribution Strategy for Object Storage Systems With Hybrid Devices", IEEE, (2017).

[2] L. shea, "Development of an Automated Storage and Retrieval System in a Dynamic Knowledge Environment", (2007).

[3] S. Sheikh-Bahaei, "Discrete Event Modeling, Simulation and Control with Application to Sensor Based Intelligent Mobile Robotics", M.Sc. Thesis, University of New Mexico, December, (2003).

[4] M. Pakdaman and M. Sanaatiyan, "Design and Implementation of Line Follower Robot ", Second International Conference on Computer and Electrical Engineering, pp.585-590, (2009).

[5] S. E. Marjani and A. Vosoughinia, "Technical Report of Building a Line Follower Robot|", International Conference on Electronics and Information Engineering ICEIE, (2010).

[6] I. AL-Taharwa, A. Sheta and M. Al-Weshah, " A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment", Journal of Computer Science 4 (4), pp. 341-344, (2008)

[7] J. Gagliardi, J. Renaud, and A. Ruiz, "On sequencing policies for unit-load automated storage and retrieval systems," International Journal of Production Research, vol. 52, pp. 1090-1099, (2014).

[8] Guest Editorial, "A revolution in the warehouse: A retrospective on KIVA systems and the grand challenges ahead", IEEE Transactions on Automation Science and Engineering, pp. 638-639, Vol.9, No.4, (2012).

[9] E. Guizzo,"Three engineers, hundreds of robots, one warehouse", IEEE Spectrum, Vol.45, No.7, pp.26-34, (2008).

[10] D. Andrea, R. Guest, "Editorial: Can Drones Deliver", IEEE Transactions on Automation Science and Engineering, L. 11, NO. 3, pp. 647-648, June (2014).

[11] K. Harada, T. Tsuji, K. Nagata, N. Yamanobe, H. Onda, "Validating an object placement planner for robotic pick-and-place tasks", Robotics and Autonomous Systems, Vol.62, No. 10, pp.1463-1477, October (2014).

[12] R. Diankov and J. Kuffner, "Open RAVE: A planning architecture for autonomous robotics, CMU, Robotics Institute", Tech. Rep. CMU-RI-TR-08-34, 79, July (2008).

[13] A. A. Ali, A. T. Rashid, M. Frasca and L. Fortuna, "An algorithm for multi-robot collision-free navigation based on shortest distance", Robotics and Autonomous Systems, Vol.75, pp.119-128, 2016.

[14] A. T. Rashid, A. A. Ali, M. Frasca and L. Fortuna, " Multi-robot collision-free navigation based on reciprocal orientation", Robotics and Autonomous Systems, Vol.60, pp.1221-1230, 2012.

[15] J. H. Hwang, R. C. Arkin, and D. S. Kwon,"Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control", Conference on Intelligent Robots and Systems, October 2003.

[16] F. R. Ali and A. T. Rashid, "Software implementation of objects store system using line follower robots ", Second Al-Sadiq International Conference on Multidisciplinary in IT and Communication Science and Applications (2nd-AIC-MITC'17), 2017.

[17] F. R. Ali and A. T. Rashid," Design and implementation of static and dynamic objects store systems using line follower robots ", International Conference on Advances in Sustainable Engineering and Applications (ICASEA'18), 2018.

[18]