DOI: 10.24237/djes.2019.12107 eISSN 2616-6909

Obstacle Avoidance Techniques for Robot Path Planning

Arif A. AL-Qassar, Ali N. Abdulnabi
[1] *Department of Control and system Engineering, University, of Technology*
alinajem18@yahoo.com , arifqassar@yahoo.com

Abstract

This paper presents a collision-free path planning approaches based on Bézier curve and A-star algorithm for robot manipulator system. The main problem of this work is to finding a feasible collision path planning from initial point to final point to transport the robot arm from the preliminary to the very last within the presence of obstacles, a sequence of joint angles alongside the path have to be determined. To solve this problem several algorithms have been presented among which it can be mention such as Bug algorithms, A-Star algorithms, potential field algorithms, Bézier curve algorithm and intelligent algorithms. In this paper obstacle avoidance algorithms were proposed Bézier and A-Star algorithms, through theoretical studies and simulations with several different cases, it's found verify the effectiveness of the methods suggested. It's founded the Bézier algorithm is smoothing accurate, and effective as compare with the A-star algorithm, but A-star is near to shortest and optimal path to free collision avoidance. The time taken and the elapsed time to traverse from its starting position and to reach the goal are recorded the tabulated results show that the elapsed time with different cases to traverse from the start location to destination using A-star Algorithm is much less as compared to the time taken by the robot using Bézier Algorithm to trace the same path. The robot used was the Lab-Volt of 5DOF Servo Robot System Model 5250 (RoboCIM5250).

Keywords: robot path planning, obstacle avoidance, Bézier curve, A-star algorithm

Paper History:

(Received: 15/6/2017; Accepted: 22/11/2017)

1. Introduction

Path planning was developed in several fields, like artificial intelligence, robotics or management principle. this is often why each individual uses terribly own definition of this term in robotic path planning with considerations with bother a show to move a robot from one point to each other point. With the advances in robot course planning with to boot includes several complications like uncertainties a few of robots or dynamics. In robotic science, path planning with manner a glance for a series of logical moves that process associate degree initial robot state right into a most popular purpose nation[1]. A manipulator without sensors has no ability to avoid obstacles in its workspace and it has to be taught every point on its trajectory, so that the arm may be free from collision as the arm moves along desired path from start point to a goal point [2]. Bézier curves have several advantages for geometric modeling. The first and last control points are coincident with the endpoints of the curve segment. The curve is also tangent to the first and last edges of the control polygon and the curve generally follows the shape of the control polygon, making it intuitive to modify Bézier curves can also be strung together, providing automatic continuity between the endpoint of one Bézier curve and the starting point of another Bézier curve [3]. Some properties of bézier curves are the start line and finishing point of bezier curve superpose the ones of relative function polygon, specifically superpose start line and finishing point with the object path. Both first derivative and second derivative of Bezier are consecutive, which means that Bezier is smooth and easy [4]. A-star is one of the many seek algorithms that takes enter, evaluates some of feasible paths and returns a solution. In computer technology, a-star is a computer algorithm this is extensively used in path locating and graph traversal, the procedure of plotting an efficaciously traversal path among points, called d nodes. The A-star set of rules combines features of uniform-price seek and pure heuristic seek to successfully compute gold standard solutions. A-star achieves higher overall performance (with respect to time) by using heuristics [5]. A-star algorithm is a heuristic search algorithm, its evaluation function determines the searching direction and affects the searching efficiency and searching results [6]. A-star is a set of rules has characteristic of gradual searching for pace and might without problem fall into the failed search state while entice obstacles are met within the unknown environments [7]. Number of studies has been conducted about path planning obstacle avoidance techniques. Jile Jiao, et.al (2013) [8] studied a collision-free path making plans technique for the mobile manipulator supported totally on Bézier curves. Simulation effects show that the Bézier curve based totally path planning May also guide the mobile manipulator to method the item smoothly without collisions with environmental boundaries. Xinping et.al. (2015) [9] offered a unique path-smoothing set of rules to correctly generate a smooth path for a car-like vehicle working in a large medical facility environment. With the proposed set algorithm, a smooth direction pleasant curvature continuity and maximum curvature constraints can be acquired. Jayasinghe and Athauda (2016) [10] proposed a collision avoidance algorithm has been provided the use of bézier curves. The proposed algorithm is a set of rules functions are validated to be well suited with the direction planning system. For that reason powerful direction making plans algorithm is usually recommended utilizing Bézier curve residences and demonstrated applicable in actual time with use of shaped simulation framework. Zhanying and Ziping (2014) [11] suggested multiple mobile robots path designing algorithm program supported A-star and Dijkstra algorithm program is projected. Additionally, for the requirements of sensible engineering, A-star and Dijkstra algorithmic program has been optimized severally. Both two algorithms are optimized too. Lin wang et.al. (2015) [12] An improved A-star set of rules is proposed, which introduces elements of turning, and part elimination based on the progressed A-star set of rules is followed to find shortest path problem. Priyanka and Velappa (2016) [13] Presented an amended new algorithm for path planning of a mobile robot for collision avoidance using optimization techniques. A modified version of the existing A-star algorithm is used to guide the robot in a complex unknown environment with static obstacles to reach the target without any collision.

2.Mathematical Formulation

2.1 Bézier Curve

Bézier curve is outlined by a series of two or additional management points. The management points frame what's referred to as the control polygonal shape. Linear segments that connect the ordered series of management points from the control polygonal shape. Bézier curves have an associated degree. The degree, L, of a Bézier curve is:

$$L = n - 1 \qquad (1)$$

Where n is the number of control points. Thus, a 1st degree Bézier curve has two control points; a 2nd degree Bézier curve has three control points, etc. The Bézier curve passes through the beginning and final control points but not the intermediate nodes that outline the beginning and also the final orientation and also the form of the curve [14]. Given a set of control points P0, P1,…Pk, the corresponding Bézier curve is defined as:

$$\mathrm{P}(\lambda) = \sum_{i=0}^{k} \left(B_{i}^{k}\right)(\lambda)\ Pi \quad 0 \le \lambda \le 1 \qquad (2)$$

Where $\left(B_{i}^{k}\right)$ is a Bernstein polynomial and Pi stands for the ith vectors of the control point, the Bernstein polynomials, which are the basis functions in the Bézier curve expression, are defined as:

$$B_{i}^{k}(\lambda) = \binom{k}{i} \lambda^{i} (1-\lambda)^{k-i} \ , i = 0,1,..,k \qquad (3)$$

Where

$$\binom{k}{i} = \frac{k!}{i!\,(k-i)!} \qquad (4)$$

The following conventions apply if u equal zero, then u^{i} =1 and 0! = 1 if there are (k+1) vertices, then the function $B_{i}^{k}(\boldsymbol{\lambda})$ yields a k^{th} degree polynomial.

Expanding Equation 3 for curves defined by three points produces the following polynomial forms: For three control points (k=2), the basis functions are:

$$B_{0}^{2} = (1-\lambda)^{2} \qquad (5)$$

$$B_{1}^{2} = 2\lambda(1-\lambda) \qquad (6)$$

$$B_{2}^{2} = \lambda^{2} \qquad (7)$$

And the curve basis function is:

$$P(\lambda) = (1-\lambda)^{2}P0 + 2\lambda(1-\lambda)P1 + \lambda^{2}P2 \quad (8)$$

$$P(\lambda) = [(1-\lambda)^{2} * 2\lambda(1-\lambda) * \lambda^{2}] * (P0, P1, P2)^{T} \qquad (9)$$

$$P(\lambda) = (\lambda^2, \lambda, 1)\begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} P0 \\ P1 \\ P2 \end{bmatrix} \quad (10)$$

When k=1, 2, 3 and 4 in Equation 2 applying the shape of the Bézier curve is completely defined by the location of the control points as showing in Figure 1. The control polygons are shown as lines connecting the control points [14].

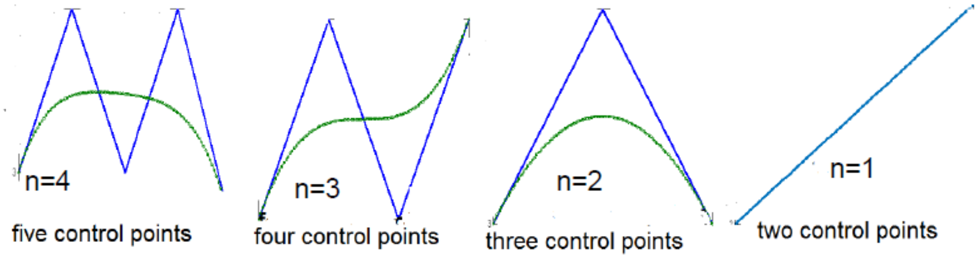


Figure 1: examples of Bezier curve

3. A-Star Algorithm

A-star algorithm is one in all the foremost effective path planning approach which can be applied on metric or topological configuration space. This program uses combination of heuristic looking out and searching out supported the shortest path. This technique uses particularly an evaluation function:

$$f(k) = h(k) + g(k) \quad (11)$$

Where $g(k)$ is that the actual cost from the beginning node to the node k (i.e. the value finding the best path), $h(k)$ is that the estimate of the worth of most helpful direction from node k to the target node, that depends on the heuristic records of the trouble regions to provide guidance and selection to the elevated nodes of the list OPEN, [7].

Advantage of this algorithm is that the distances used as a criterion may be adopted, changed or some other distance may be delivered. This offers a wide range of changes of this simple precept [1]. Applying the A-star algorithm in path searching issues is specially due to the information of the problem, solving procedure in addition to the obtained solution, to look for some records associated with and conducive to problem-solving, it is regarded because the heuristic statistics of the solution. Special from many other path searching algorithms, it does no longer traverse the whole map and all of the nodes, the looking gadget of the A-star set of guidelines is as follows:

1. To mark the start node and make bigger the unmarked subsequent.
2. To compute the assessment feature worth for each later node, to kind them to keep with the worth of the assessment characteristic and to select out and mark the node of the stripped analysis characteristic price.
3. It's essential to rent the above processing steps for it and to record the shortest direction if the present node isn't the target node.

The essential step within the a-megastar set of rules is to select the assessment characteristic. it's going to guide the searching closer to the most promising path to be in search of the shortest and highest quality path if the assessment function is selected well [7].

4. Path Planning of Obstacle Description

There are a lots of paths to the target point in the related area. These paths may be connected by multi-lines or smooth curves. Using Bézier and A-star to describe path feasible. The obstacles of this work may have polygonal or spherical or any other shape. Every point on the robot has to be located outside the cubic obstacle to ensure a collision free trajectory with that obstacle. However, it is not particularly desirable to have the manipulator pass very close to the obstacle boundaries, and thus the smallest cube which bounds an original obstacle has been used in this work to approximate the obstacle.

5. Generating Path Planning Based on Bézier And A- Star Algorithm

The developed Path designing supported Bézier curve methodology to transport the robotic arm through variety of intermediate points to achieve the required goal point is approached within the following way:

1. Enter Cartesian coordinates of initial, final, and obstacles points.
2. Apply Bézier curve formula in Equation 2

3. Partitioning the generated curve into sufficient intermediate nodes.
4. Convert the Cartesian space of each generated intermediate nodes into joint space by applying inverse kinematics analysis at these points.
5. Move the robotic arm through the generated joints spaces from the initial point towered the final point.

A-star algorithm that to be generate path planning obstacles avoidance from start towards goal points in the following steps:

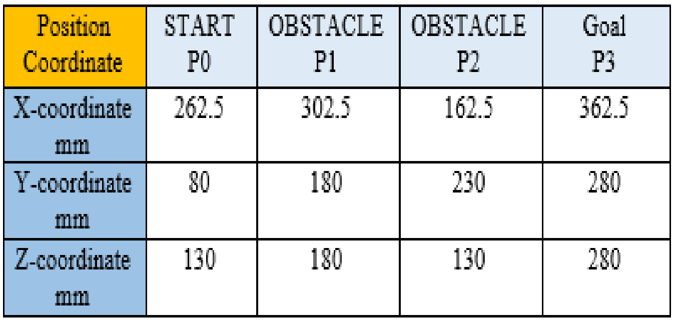
1. Put the start node on the list OPEN and calculate the cost function.
2. Remove from the List OPEN the node with the tiniest value operate and place it on CLOSED. This can be the node n. (Incase 2 or additional nodes have the value operate, indiscriminately resolve ties. If one in all the nodes is that the goal node, then choose the goal node).
3. If n is that the goal node then terminate the guideline and use the suggestions that might get the solution direction. Otherwise, retain.
4. Calculate all the successor nodes of n and compute the cost characteristic for each successor not on list CLOSED.
5. Companion with each successor not on listing OPEN or CLOSED the fee calculated and place these at the listing OPEN, setting hints to n (n is that the parent node).
6. Companion with any successors already on open the smaller of the fee worth's simply calculated and therefore the previous fee value.

6. Simulation Results And Discussion

According to the mathematical formula of Bézier curve and therefore the coordinate of the control points. Several cases that to be implemented to generate Bezier curve:-

In the first case, two obstacles in this case four nodes [start, obstacle one, obstacle two and goal] with coordinates as shown in the Table 1, have been modeled to generate third order Bézier curve n−3 according to the Equation 2 and represented as illustrated in Figure 2. Using Matlab software.

Table 1 coordinate of two obstacles.



Position Coordinate	START P0	OBSTACLE P1	OBSTACLE P2	Goal P3
X-coordinate mm	262.5	302.5	162.5	362.5
Y-coordinate mm	80	180	230	280
Z-coordinate mm	130	180	130	280

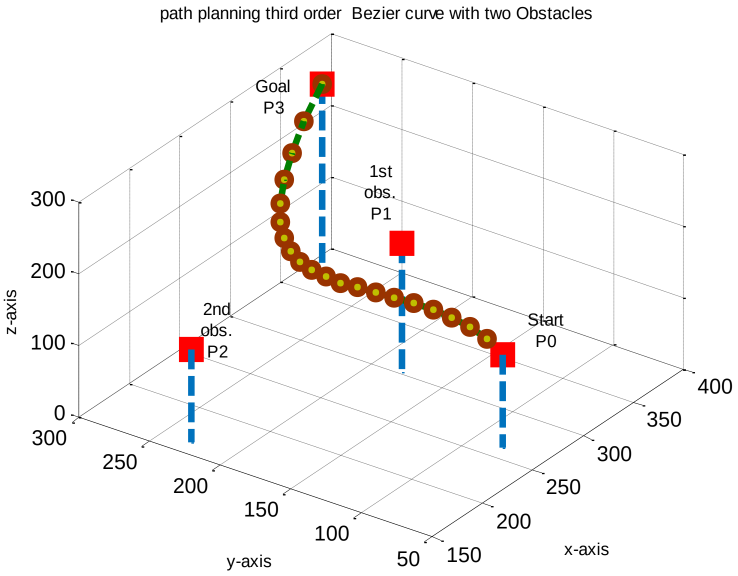


Figure 2: path planning using Bézier curve with two obstacles (case-1)

In the second case, third obstacles in this case five nodes [start, obstacle one, obstacle two, obstacle three and goal] with coordinates as shown in the Table 2, have been modeled to generate forth order Bézier curve, n−4 according to the Equation 2 and represented as illustrated in Figure 3. Using Matlab software.

Table 2 coordinate of three obstacles.

Position Coordinate	START P0	OBSTACLE P1	OBSTACLE P2	OBSTACLE P3	Goal P4
X-coordinate mm	180	180	330	330	180
Y-coordinate mm	80	230	280	330	380
Z-coordinate mm	130	330	230	430	430

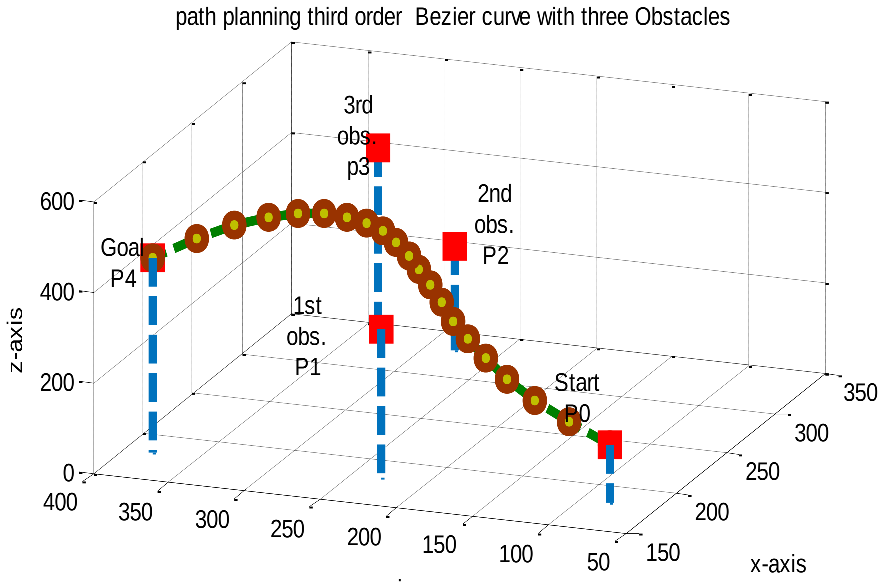


Figure 3: path planning using Bézier curve with three obstacles (case-2)

In the third case, four obstacles in this case six nodes [start, obstacle one, obstacle two, obstacle three, obstacle four and goal] with coordinates as shown in the Table 3, have been modeled to generate fifth order Bézier curve, n−5 according to the Equation 2 and represented as illustrated in Figure 4. Using Matlab software.

Table 3 coordinate of four obstacles.

Position Coordinate	START P0	OBSTACLE P1	OBSTACLE P2	OBSTACLE P3	OBSTACLE P4	Goal P5
X-coordinate mm	330	80	230	330	130	230
Y-coordinate mm	80	130	230	280	330	380
Z-coordinate mm	110	230	330	380	280	255

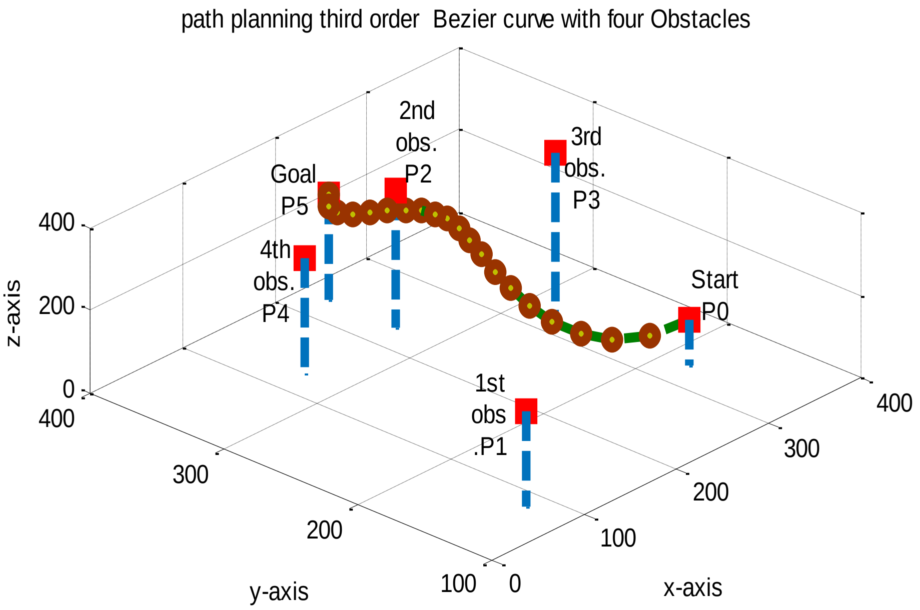


Figure 4: path planning using Bézier curve with four obstacles (case-3)

According to A-star algorithm the environment of work is unknown and static obstacles are present and the robot is guided to track the path to reach the target. Three cases have been implemented:

Case-1 shows one of the tests of A-star Algorithm Figure 5 shows the path planning of the robot using the A-star algorithm here the obstacles are static and the positions of the obstacles and the target are fixed. The robot is initially positioned at coordinate (0, 0), and the obstacle-1 starting at (20, 40) and extending upto (40, 20). Similarly obstacle-2 starts at (20, 50) and extends upto (60, 50) and the target is located at (70, 60). Once the robot starts, it traverses through the optimal path determined by the algorithm and reaches the target. The time taken for the robot to start and reach the target in the Case is measured to be 1.922 sec and the elapsed time is 0.522 sec. Figure 6 shows the contour diagram of the path followed by the robot through the environment and reach the target.

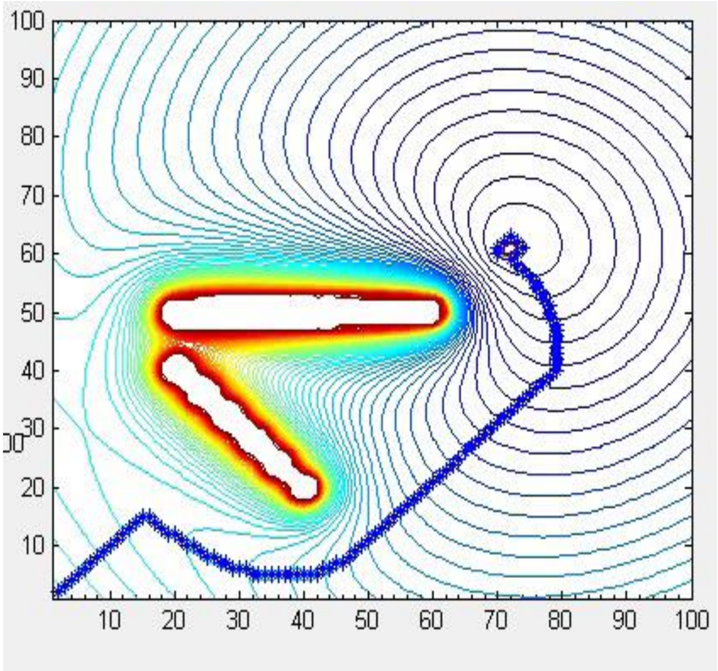


Figure 5: Path traversed using A-star Algorithm (case-1)

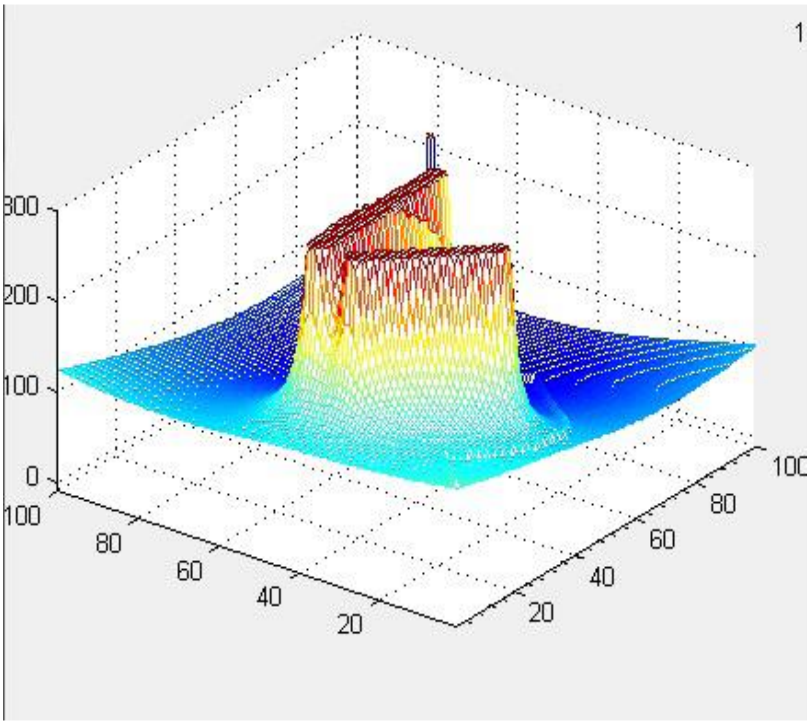


Figure 6: Contour Diagram of the Path traversed by the robot using the A-star Algorithm (case-1)

In Case-2, Figure 7 shows the trajectory traversed using the A-star algorithm in 10 x 10 grid environment. In this case, the positions of the target and the obstacles are fixed and the obstacles are assumed to be static. The robot is positioned at coordinate (0, 0), obstacle-1 is placed starting at (20, 10) and ending at (20, 50), obstacle-2 at (20, 10) and ending at (50, 10) and the target is located at (40, 30). Once the robot starts moving, it traverses through the optimal path found by the A-star Algorithm and reaches the target. The time traversed by the robot is 3.334 sec and the elapsed time is 0.906 sec. Figure 8 shows the contour diagram of the path followed by the robot through the environment and reach the target.

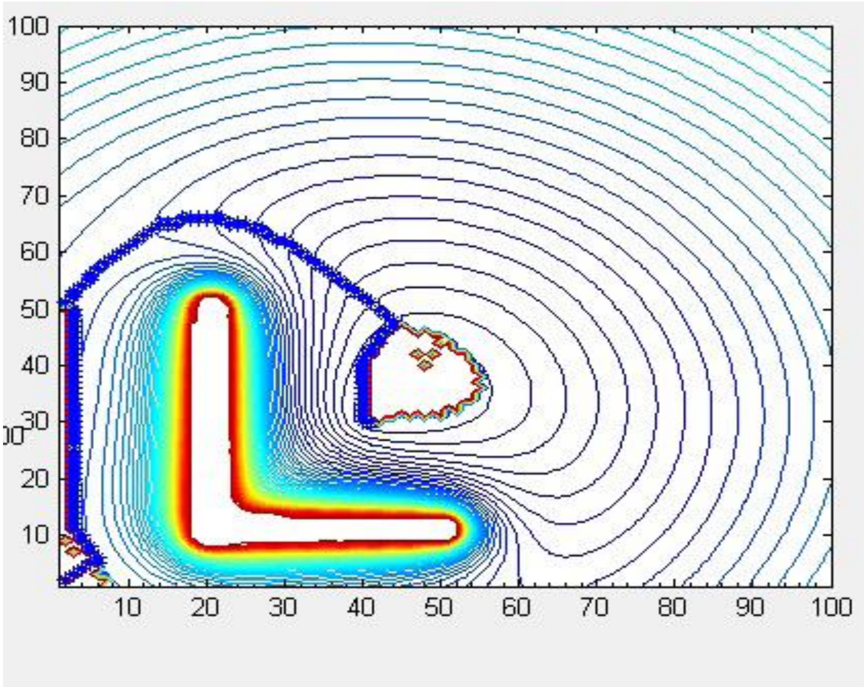


Figure 7: Path traversed using the A-star Algorithm (case-2)

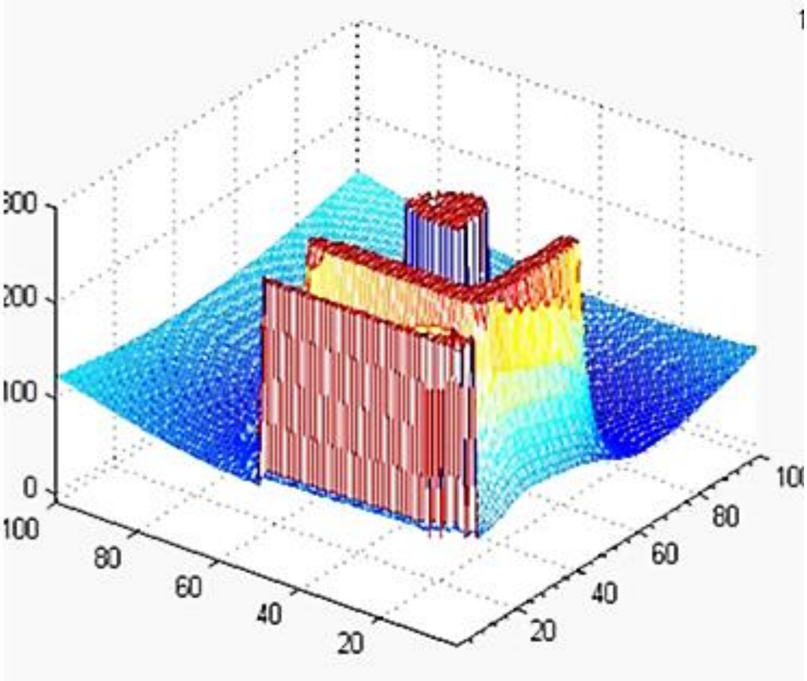


Figure 8: Contour Diagram of the Path traversed by the robot using the A-star Algorithm (case-2)

In case-3, Figure 9 shows the trajectory traversed using the A-star algorithm In this Case, the positions of the target and the obstacles are fixed and the obstacles are assumed to be static. The robot is positioned at coordinate (0, 0), obstacle-1 at (30, 60) to (50, 10), and the target at (60, 40). The robot starts traversing through the optimal path calculated by the A-star Algorithm and reaches the target. The time traversed to reach the goal is 4.671 sec and the elapsed time is 1.671 sec. Figure 10 shows the contour diagram of the path followed by the robot through the environment and reach the target the path planning of the robot that stated in description part for both techniques Bézier curve and A-star star are described and the results are discussed. In the case one the obstacles are static and the target is fixed. Once the simulation for the Case one starts, the robot traverses from its starting point through the path determined by both the A-star and Bézier Algorithms and reach the target.

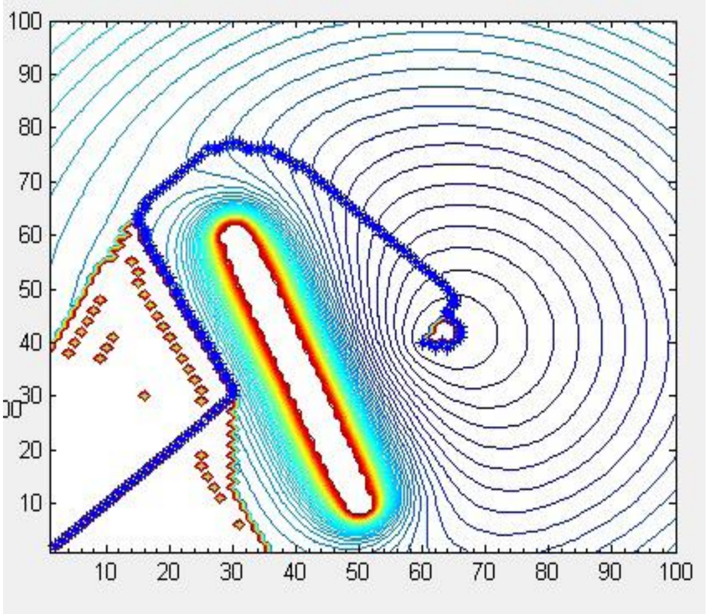


Figure 9: Path traversed using the A-star Algorithm (case-3)

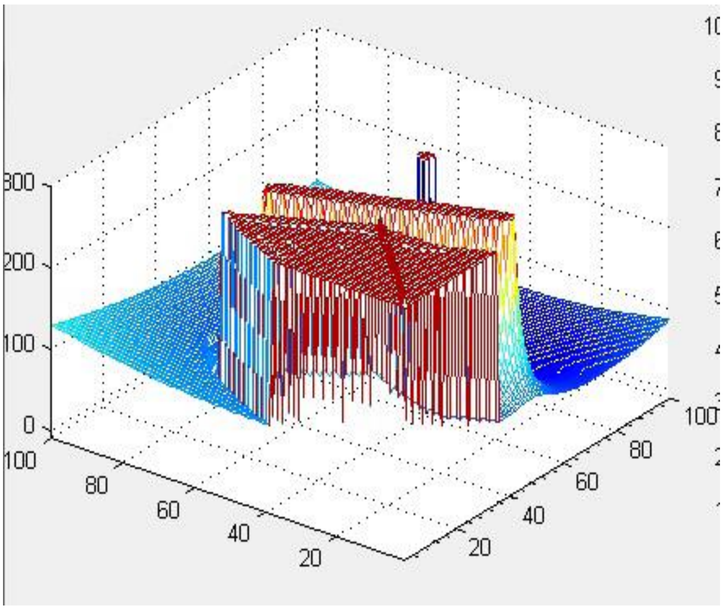
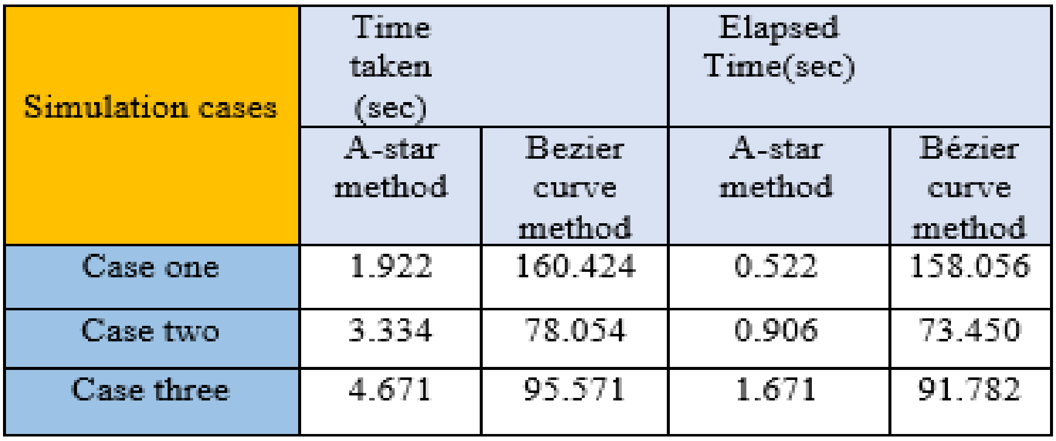


Figure 10: Contour Diagram of the Path traversed by the robot using the A-star Algorithm (case-3)

The time taken and the elapsed time to traverse from its starting position and to reach the goal are recorded and given in Table 4. Similarly the scenario for the Case two and three is established with static obstacles and fixed targets. Table 4 shows the time taken and the elapsed time traversed from starting position to the goal in Case1, 2 and 3 using the A-star Algorithm and the Bezier Algorithm.

Table 4 Time taken and elapsed time traversed by the robot using the A-star algorithm and the Bezier curve algorithm to reach the target.



Simulation cases	Time taken (sec)		Elapsed Time(sec)	
	A-star method	Bezier curve method	A-star method	Bézier curve method
Case one	1.922	160.424	0.522	158.056
Case two	3.334	78.054	0.906	73.450
Case three	4.671	95.571	1.671	91.782

7. Conclusion

By applying Bézier techniques, curves with different orders and proposed algorithm A-star to reach the target. In this paper it's expect that the solution developed will make the use of Bézier techniques more useful in applications of robot path planning in known environments (obstacles) without collision. While the proposed A-star algorithm more useful in specific applications with unknown and static obstacles. Time taken by move the robot to traverse from the initial position to the target using Bézier Algorithm is more greater as compared with time taken to traverse from the same position to the target using A-star Algorithm.is It is evident that the results obtained using A-star Algorithm are shortest near optimal than the results obtained using the Bézier Algorithm. From Table 4, the tabulated results show that the elapsed time to traverse from the start location to destination using A-star Algorithm is much less as compared to the time taken by the robot using Bézier Algorithm to trace the same path. By procedure of Bézier and A-star techniques with several different cases, it's found that the Bézier technique is an efficient, accurate, smooth and as compared with A-star technique. Based on simulation results, the following can be highlighted:

1. Make the algorithm in an unknown environment with dynamic obstacles present and implement the same in real time.
2. Generating obstacle avoidance system for the robot using image processing by computer vision technique.
3. Finding free collision path for the robot in case of fixed obstacles using 3 IR sensors.

Acknowledgment:

This work was supported by University of technology control and system Engineering Department, which is gratefully acknowledged.

References

[1].Frantisek D., Andrej B., Martin K., Peter B., Martin F. and Tomas F., Path planning with modified A star algorithm for a mobile robot, Procedia Engineering, 96, (2014), pp.59 – 69.

[2].Namgung I., Application of Quadratic Algebraic Curve for 2D Collision-Free Path Planning and Path Space Construction, International Journal of Control, Automation, and Systems, 2(1), (2004), pp.107-117.

[3].Ji-wung C., Renwick E. and Gabriel H., Smooth path generation based on Bezier curves fot autonomous vehicles, proceeding of the world congress on engineering and computer science, San Francisco, Vol.II, (2009),

[4].Yang L., Luo Z., Tang Z. and Lv W., Path Planning Algorithm for Mobile Robot Obstacle Avoidance Adopting Bezier Curve Based on Genetic Algorithm, Chinese Control and Decision Conference (CCDC), 80, (2008), pp.3286 –3289.

[5].Harika R., PATH FINDING - Dijkstra’s and A* Algorithm’s, computer science and electronic engineering, 10, (2013), pp.1– 15.

[6].Liping C., Chuanxi L. and Bo Y., Improved Hierarchical A-star Algorithm for Optimal Parking Path Planning of the Large Parking Lot, International Conference on Information and Automation, 2, (2014), pp.695 – 698.

[7].Junfeng Y., Chao L., Xiaobiao X., Andy J. W. and Cheng H., Path Planning for Virtual Human Motion Using Improved A* Algorithm, International Conference on Information Technology, 10, (2010), pp.1154 – 1158.

[8].Jile J., Zhiqiang C., Peng Z., Xilong L. and Min T., International Conference on Robotics and Biomimetics, 3, (2013), pp.1864 –1868.

[9].Xinping B., Hu S., Wei Z. and Peng W., Curvature Continuous Path Smoothing Based on Cubic Bezier Curves for Car-Like Vehicles, IEEE Conference on Robotics and Biomimetic, (2015), pp.1453 – 1458.

[10]. Jayasinghe J. A. and Athauda A. M., Smooth Trajectory Generation Algorithm for an Unmanned Aerial Vehicle (UAV) Under Dynamic Constraints, (2016).

[11]. Zhanying Z. and Ziping Z.,A, Mobile Robots Path planning Algorithm Based on A-star and Dijkstra Algorithm, International Journal of Smart Home, 8, (3), (2014), pp.75-86.

[12]. Chunbao W.,Lin W. and Jian Q., Path Planning of Automated Guided Vehicles Based on Improved A-StarAlgorithm, International Conference on Information and Automation, 6, 2015, pp.2071 – 2076.

[13]. Priyanka S. and Velappa G., (2016), Path Planning of a Mobile Robot using Amended A-Star Algorithm, International Science Press, 9, (37) , pp. 489-502.

[14]. Renwick E. C., Gabriel H. E., (2007), Continuous Curvature Path Generation Based on Bezier Curves for Autonomou Vehicles, International Journal of Applied Mathematics, 2,(40), pp.91 – 101.