American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Optimal Path Planning Obstacle Avoidance of Robot Manipulator System using Bézier Curve

Arif A. AL-Qassar[a*], Ali N. Abdulnabi[b]

[a,b]*University of Technology, Control and Systems Engineering Department*

[a]*Email: arifqassar@yahoo.com*

[b]*Email: alinajem18@yahoo.com*

Abstract

One of the problem that faced by engineers in most automated factories that require the need to move things from one place to another in an automated space with obstacles on its way the shortest route and the least time it takes to reach the goal. This paper presents an optimal path planning of 5DOF Lab-Volt 5250 robot manipulator joints and gripper to move from the given start point to the desired goal point without any collision with the obstacles whose boundaries are enveloped by a spherical shape, the size and the height of the obstacle is taken into account. The path planning approach presented is suggested in the robot joint space by using Bézier curve technique. The particle swarm optimization PSO method is used to get the optimal path with the shortest distance and the least time to move the end-effector from the initial point to the final point without hitting any obstacles which exist in the robot environment. This work is not limited to theoretical studies or simulations, but several experiments cases were tested in different situations in a static environment known to test the robot's arm's ability to reach the desired target without hitting any obstacles with the shortest distance and least time.

Keywords: Path Planning; obstacle avoidance; Bézier curve; particle swarm optimization.

1. Introduction

A robot is a designed to performance of a variety of tasks by reprogrammable multifunctional process to movement parts, tools, material, or specific devices through variable programmed motions [1]. These days most factories automated manufacturing and productions, are implemented by specific machines designed to perform duties automatic in a manufacturing and production process.

----

* Corresponding author.

The stiffness and the usually high cost of these machines, frequently recognized to tough mechanization systems, have led to a broad-based interest in the use of robots able to performing a variety of manufacturing functions in a greater flexible operating environment. The arm of the robotic is the most utilized part generally. In addition, it is used in many applications in manufacturing and production. For instance, welding manipulators are use as spot-welding robots in manufacturing automobile, spray nozzles and paint with a need for improved production to give us products of uniform satisfactory, therefore factories are turning towards computer-based automation [2]. Robotic arm manipulator is one of the motivating disciplines in business and academic applications, and an essential branch to control sciences because of its intelligent aspects, nonlinear characteristics, and its actual time implementation. It became evolved to enhance human's work which includes in the production of heavy materials, and unpredictable environments. Regardless of the type of task robot manipulator can be provided with, robot overall performance measures the high best and massive amount of labor that the robotic can do in the desired time and vicinity. Robot manipulator has immeasurable responsibilities, so it's far designed to be flexible in general motions to transport things from one position to another with easy movement to keep away from sharp jolt inside the robotic arm, those jolts may harm the arm [3]. Number of studies has been conducted about optimal path planning obstacle avoidance techniques. Lars and Brian in 2006, proposed a new, complete algorithm for manipulator path planning with obstacles. By posing the problem as a Disjunctive Program, existing constrained optimization methods were used to generate optimal trajectories [4]. The optimum collision-free path planning of a 6-DOF robot manipulator using multiple optimization criteria was developed by Mitsi and his colleagues in 2014 The optimum collision free path planning was solved by a hybrid method that combines a genetic algorithm(GA), a quasi-Newton algorithm and a constraints handling method, using a various constraints and multi-objective function [5].The optimal path planning of a robot manipulator, which was used in a surgery, was obtained by Muhammad and his colleagues in 2015, who used Genetic Algorithm (GA) to solve the optimal path planning problem in environment with variable number of generations. Genetic Algorithm (GA) has determined the way point accurately. It can help to choose the close points or the optimal points and ignore unworkable points [6]. Andre and his colleagues in 2016, presented a decision algorithm that can be online implemented and is able to find an optimal minimum time law along the path. This law minimizes the time required to complete the path and at the same time is consistent with constraints, both at kinematic and dynamic levels [7]. Amr and his colleagues in 2017, presented a motion planning algorithm for generating optimal collision-free paths for robotic vehicle with two trailers moving autonomously. The suggested algorithm is depend on a combination between two approaches artificial potential field method and optimal control theory [8].

2. Path Planning with Obstacles Avoidance Based on Bezier Curve

Generated path planning of obstacles by using a Bézier curve technique, where control points of Bézier curve have been suggested to generate feasible path while the generated Bézier curve can be divided into intermediate points to guide the robot arm to move from start position to goal position. A Bézier curve is defined by a series of two or more control points which are known as control polygon. Linear sections are connecting the ordered series of control points from the control polygon [9]. The degree, n, of a Bézier curve is:

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

Where m is the number of control points. Thus, a Bézier curve with two control points is of first degree, a Bézier curve with three control points is of second degree and so on. Examples of Bezier curves are shown in Fig. (1). the control polygon is shown as lines connecting the control points. By changing or moving theses control points, the shape of the curve is globally changed in a mathematically manner [10].

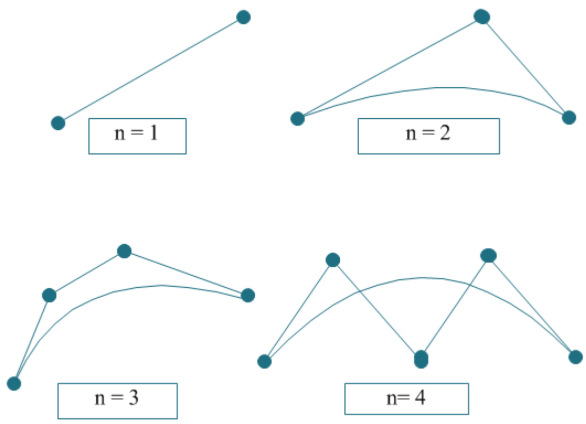

Figure 1: Examples of Bezier curve

In this paper the property of the Bézier curve has been used to plan collision free path planning that is the curve which passes only through the start point and the goal point of control points and does not pass through the other control points. By considering the start and the goal control points as the initial and the final positions, respectively, of the robot path while the control points are considered as obstacles. A sequence of joint angles along the path must be determined in order to move the robot arm from start to goal points with exist obstacles of the robot environment [11].

3. Mathematical Formulation of Bezier Curve

The following definition, of the typical Bézier space curve as an example is given:

$$\mathrm{P(t)} = \sum_{i=0}^{n} P_i \; B_{i,n}\,(\mathrm{t}) \qquad 0 \leq \mathrm{t} \leq 1 \qquad (2)$$

Where $B_{i,n}$ is a Bernstein polynomial and P_i stands for the i_{th} vector of the control point, the Bernstein polynomials, which are the basis functions in the Bézier curve expression, are defined as:

$$B_{i,n}\,(\mathrm{t}) = \binom{n}{i} t^i\,(1-t)^{n-i} \quad , i = 0,1,2 \ldots \ldots, n \qquad (3)$$

Where,

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

The following conventions apply if *i* and *t* equal zero, then t^i =1 and 0! = 1 if there are (n+1) vertices, then the function $B_{i,n}$ (t) yields a n^{th} degree polynomial. Most of engineering applications can be covered by a third degree formula when n=3, for four control points, the basis function are given by:

$$B_{0,3} = (1-t)^3 \tag{5}$$

$$B_{1,3} = 3t(1-t)^2 \tag{6}$$

$$B_{2,3} = 3t^2(1-t) \tag{7}$$

$$B_{3,3} = t^3 \tag{8}$$

So that:

$$\mathrm{P(t)} = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3\mathrm{t}^2(1-t) P_2 + \mathrm{t}^3 P_3 \tag{9}$$

When n=3, cubic Bézier curve, in equ. 9 is written in a matrix form as given:

$$\mathrm{P(t)} = [(1-3t+3\mathrm{t}^2-\mathrm{t}^3)(3t-6\mathrm{t}^2+3\mathrm{t}^3)(3\mathrm{t}^2-3\mathrm{t}^2)\mathrm{t}^3]\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{10}$$

$$\mathrm{P(t)} = [\mathrm{t}^3\ \mathrm{t}^2\ t\ 1]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{11}$$

4. Particle Swarm Optimization

In 1995 Kennedy and Eberhart has been proposed a Particle swarm optimization PSO and it is a new idea which widely falls below evolutionary computation techniques. Social behavior of organisms including bird flocking and fish schooling motivates researchers to look at the effect of collaboration of species when attaining their desires as a group.

Years of examine at the dynamics of chook flocking resulted within the opportunities of making use of this conduct as an optimization device. In a PSO system, a trouble space is initialized with a population of random solutions and more than one candidate solutions coexist and collaborate concurrently.

Every solution candidate, referred to as a "particle", flies within the problem space (much like the search system for meals of a fowl swarm) looking for the most optimal position to land. A particle, as time passes through, adjusts its position in line with its very own "experience", as well as according to the experience of nearby particles [12].

5. Simulation cases of optimal path planning with obstacles avoidance

The optimal path planning was carried out by relying on Bézier technique method and it is illustrated in the following steps:

These steps are used to generate the path planning depending on Bézier curve technique. Bézier chose a family of functions called Bernstein polynomials.

Step one: input Cartesian coordinates of start point, goal point and obstacles.

Step two: apply Bézier curve technique in equ. 2.

Step three: partition the generated curve into sufficient intermediate points.

Step four: convert the Cartesian space of each generated intermediate point into joint space by applying the inverse kinematic solution.

Step five: move the robot arm through the generated joints spaces from the start point toward the goal point.

Step six: used the PSO Matlab program with suitable parameters to get the best feasible path of obstacles avoidance while the robot manipulator moves from the start to the goal positions.

For all Figures in all cases, the square markers with yellow and green colors represent the start and the goal points, respectively and the other circle marker with blue color represents the obstacles, by using Matlab software program.

5.1 Path planning with obstacles avoidance: case one

According to the mathematical formulae of Bézier curve and the mathematical representation of PSO, there are three obstacles in this case, five nodes [start, obstacles, and goal] with coordinates as shown in the Table 1.

Which have been modeled to generate 4^{th}order Bezier curve .

The PSO parameters are set as: population size = 150, C_1, C_2 =1.5, $w = 1$ and the number of iterations is selected to be 200.

It was found from many simulations that the above settings for the PSO parameters were adequate. In particular, when the number of iterations was set to 200, the best optimal cost function, was obtained and any increasing in the number of iterations did not improve the convergence of the algorithm significantly. The difference between the two paths without and with applying PSO in 2D environment is illustrated in Fig. (2).

Table 1: Coordinates of five control points

Position Coordinate	Start P0	Obs. P1	Obs. P2	Obs. P3	
X-coordinate cm	22.26	38.17	35.039	50.56	54.521
Y-coordinate cm	-32.88	-23.75	-8.045	6.353	20.685
Z-coordinate cm	25.56	27.15	23.017	20.248	15.028

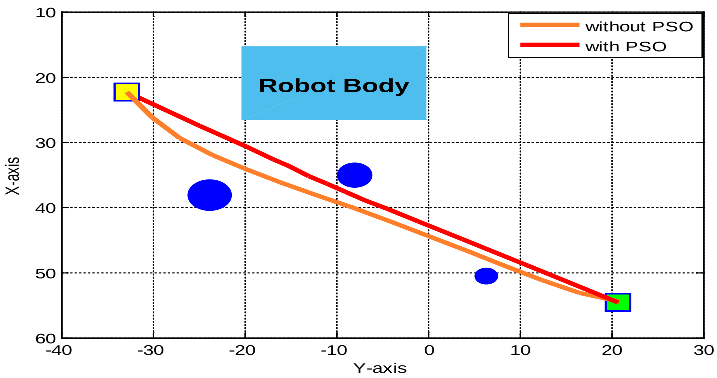

Figure 2: Difference between the two paths of obstacles avoidance without and with applying PSO case 1

The total distance of the generated curve in 3D environment without applying PSO is 64.5115 cm and the Time taken is 8.601 sec.

and the total distance of the generated curve in the same environment with applying PSO is 63.499 cm and the Time taken is 8.466 sec.

5.2 Path planning with obstacles avoidance: case two

In the second case, five obstacles including seven nodes [start, obstacles and goal] with coordinates as shown in the Table 2.

Have been modeled to generate a 6^{th} order Bézier curve and the PSO parameters are set as: population size = 150, C_1 , C_2 =1.5, $w = 1$ and the number of iterations is selected to be 200 which was very adequate in obtaining the best cost function.

The convergence of the PSO algorithm was not improved when the number of iterations was increased. The results are illustrated in Fig. (3).

Table 2: Coordinates of seven control points

Position Coordinate	Start P0	Obs. P1	Obs. P2	Obs. P3	Obs. P4	Obs. P5	Goal P6
X-coordinate Cm	45.087	38.955	24.813	35.066	23.463	33.17	21.62
Y-coordinate Cm	16.498	6.63	0.629	-7.685	-13.492	-22.671	-27.067
Z-coordinate Cm	25.399	22.956	27.242	28.305	19.815	33.756	20.614

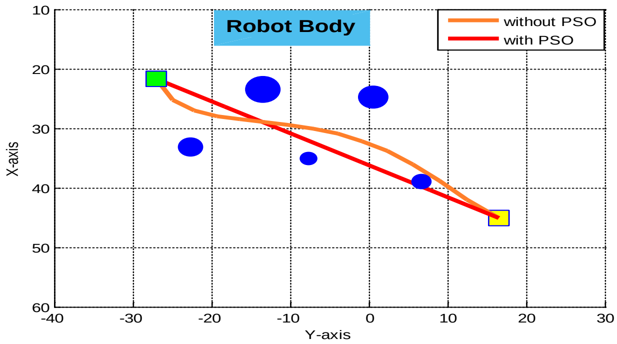

Figure 3: Difference between two paths of obstacles avoidance without and with applying PSO case 2

The total distance of the generated curve from start to goal points without applying PSO is 53.109 cm and the time taken at the same environment is 7.081 sec, and with applying PSO is 49.773 cm, and the total time taken is 6.636 sec.

6. Comparison and Discussion Simulation Cases

In the previous subsections (5.1) and (5.2) the Bézier curve was used to generate the path planning with obstacles avoidance, then the PSO was utilized to get the best cost function with minimum time and minimum distance from the start to the goal points. Table 3. Shows the comparison between path planning with the obstacle avoidance without and with applying PSO.

Table 3: shows the comparison of three simulation cases

Simulation cases	Total time taken (sec)		Total distance (cm)	
	Path planning without applying PSO	Path planning with applying PSO	Path planning without applying PSO	Path planning with applying PSO
Case one	8.601	8.466	64.511	63.499
Case two	7.081	6.636	53.109	49.773

7. Experimental setup: case one

The task is to find an optimal feasible path planning based on the Bézier curve method for the end–effecter of the robot manipulator system. To achieve this task, the Bézier curve was used to generate a suitable curve with a specific time and distance and to the get best path planning with obstacle avoidance to move the end-effector of robot manipulator from the start to the goal points via using the PSO approach. In this real environment with three obstacles five nodes [start, obstacles and goal] have been generated by using Matlab software program. From using the Robocim 5250 software program, the coordinates of the five nodes are illustrated in Table 1.The start and goal points in Matlab for all cases will be in a cubic shape with a yellow and green colors, respectively, and in the real environment, the start and goal points in all cases will be of a cubic shape having dimension as:

Length = width=height=2cm

And the three obstacles in this real case will be of spherical shapes having diameters of:

D1=6.5 cm, D2=5.5 cm, D3=4 cm

By applying the Robocim program software theta of all intermediate points along the curve can be obtained then the forward kinematic is applied to get positions of these theta and from the positions the actual curves can be plotted, as illustrated in Fig. (4).

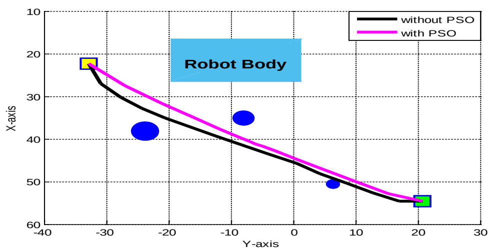

Figure 4: Two path planning of obstacles avoidance without and with applying PSO case 1

The total distance of the generated curve in this case from the Matlab program without applying PSO from start to goal points is 67.21 cm, the time taken at the same environment is 8.961 sec. and with applying PSO algorithm the total distance of the generated curve is 64.104 cm, and the total time taken is 8.547 sec. Fig. (5) Shows the path planning physical steps progress for three obstacles based on Bezier curve using the servo 5DOF robot manipulator Lab-Volt 5250 system.

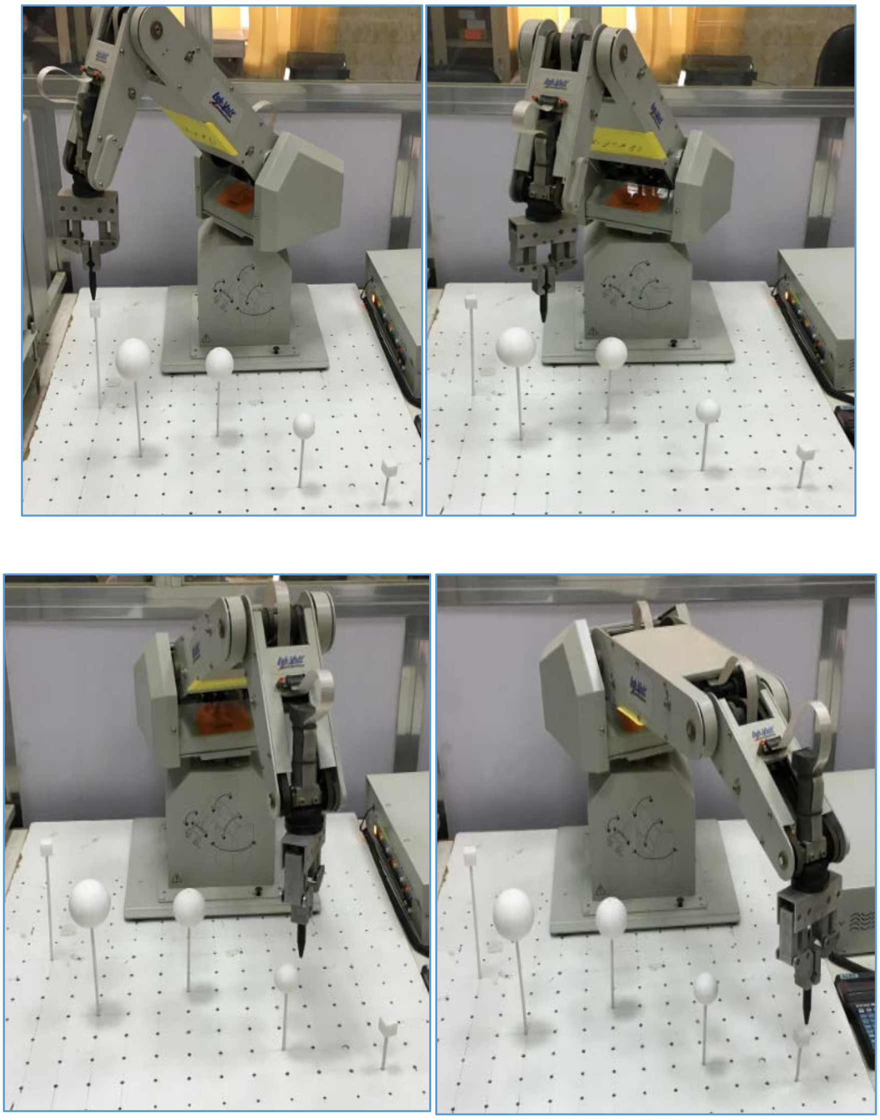

Figure 5: Physical steps case 1 of three obstacle avoidance path planning in the real environment

8. Experimental setup: case two

In this case, a real environment five obstacles environment, seven nodes [start, obstacles and goal] have been generated and the coordinates of the seven nodes are illustrated in Table 2. The five obstacles will be in spherical shape having diameters of:

D1=3.5 cm, D2=5.5 cm, D3=3 cm, D4=6 cm, D5=4.7 cm

Fig. (6) show the actual curves that are generated from applying the Robocim program which can get theta of all intermediate points along the curve then applying the forward kinematic to get the positions of these theta and from the positions actual curves can be obtained.

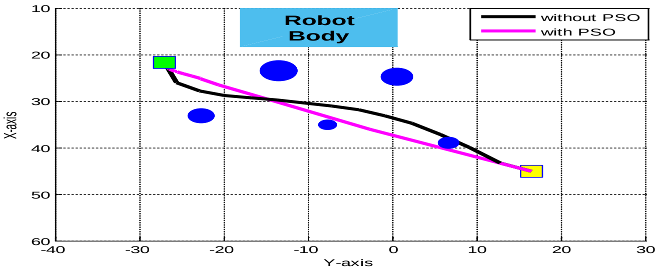

Figure 6: Two path planning of obstacles avoidance without and with applying PSO case 2

The total distance of the generated curve in this case without applying PSO from start to goal points is 55.481 cm, the time taken at the same environment is 7.397 sec. and with apply PSO algorithm the total distance of generate curve is 51.365 cm, and the total time taken is 6.848 sec. The physical steps of the path planning progress for five obstacles avoidance based on Bezier curve are illustrated in Fig. (7).

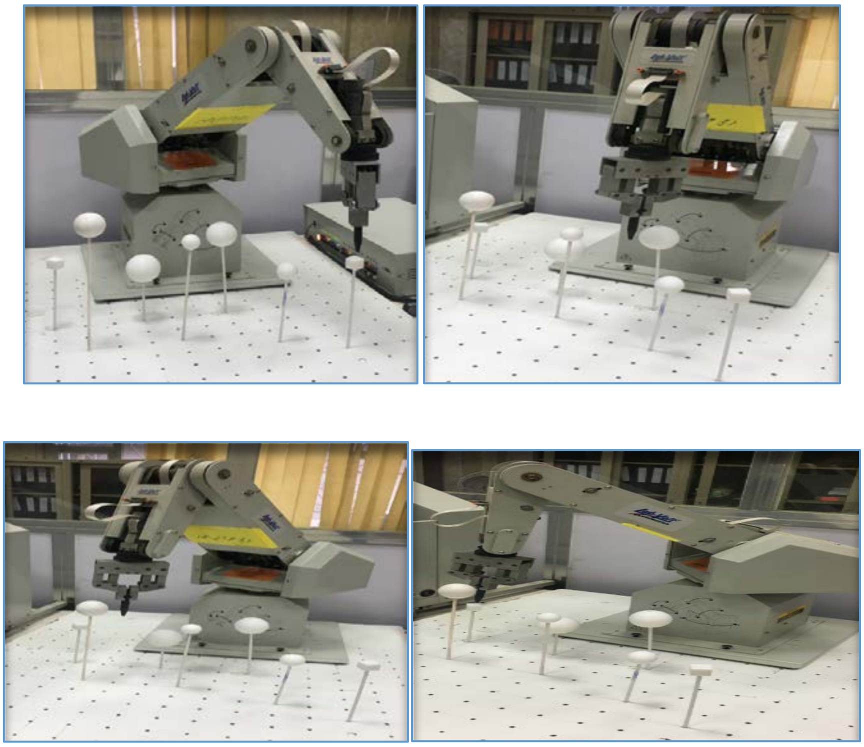

Figure 7: Physical steps case 2 of five obstacle avoidance path planning in real environment

9. Comparison and Discussion Experimental Cases

In the previous subsections (7) and (8) the Bézier curve was used to generate the path planning with obstacles avoidance, then the PSO was utilized to get the best cost function with minimum time and minimum distance from the start to the goal points. Table 4. Shows the comparison between path planning with the obstacle avoidance without and with applying PSO.

Table 4: shows the comparison of three experimental cases

Experimental cases	Total time taken (sec)		Total distance (cm)	
	Path planning without applying PSO	Path planning with applying PSO	Path planning without applying PSO	Path planning with applying PSO
Case one	8.961	8.547	67.21	64.104
Case two	7.397	6.848	55.482	51.365

10. Conclusions

In this paper, it can be calculate that: The generation of path planning by using Bézier curve should partition the curve to enough segments to generate intermediate node which the arm should follow by a sequence of joint angles to reach the goal. By testing the adopted procedure for each case, it has been found that each case take specific time and distance. The proposed algorithm for optimal path planning collision free by using Bézier technique for geometric path planning is appropriate for robots with a well-known workspace environment then the PSO is applied algorithm and this adopted algorithm to find collision free path compared to other path planning algorithms by utilizing the output of the intermediate points of Bézier curve. The main contribution of this algorithm is the decrease in the distance and time needed to map a collision free path from a given start point to final desire point when the sensor is not available in equipped robots. Converting all the obstacle shapes from nodes into circle shape at 2D space then into sphere shape at 3D space helps to represent and take into consideration the volume of all the obstacles. Segmentation of Bézier curve into two or several segments gave the path more flexibility by controlling each segment on the curve separately. By testing the adopted proposed algorithms of Bézier curve and PSO with several different cases, it was found that there is a difference between the cases without and with applying the PSO in term of achieving of less time and shorter distance with optimization ration for these cases are (1.569% and 6.281%)respectively.

References

[1] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, "Robot Modeling and Control", 1st Edition, John Wiley & Sons, 2005.

[2] Nof S," Handbook of Industrial Robotics", John Wiley & Sons, Inc, USA 1999.

[3] J. Angeles, "Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms", 2nd Edition, Springer, 2003.

[4] L. Blackmore and B. Williams, “Optimal Manipulator Path Planning With Obstacles Using Disjunctive Programming,” American Control Conference, no. 1, pp. 4–6, 2006.

[5] S. Mitsi, K.-D. Bouzakis, D. Sagris, and G. Mansour, “Robot Path Planning Optimization, Free of Collisions, Using A Hybrid Algorithm,” no. October, pp. 1–3, 2008.

[6] Z. D. Hussein, M. Z. Khalifa, I. S. Kareem, “Optimize Path Planning for Medical Robot in Iraqi Hospitals,” Engineering and Technology Journal, vol. 33, no. 5, pp. 1009–1022, 2015.

[7] A. Casalino, A. M. Zanchettin, and P. Rocco, “Online Planning Of Optimal Trajectories on Assigned Paths with Dynamic Constraints for Robot Manipulators,” IEEE International Conference on Intelligent Robots and Systems (IROS), vol. 2016–November, pp. 979–985, 2016.

[8] A. Mohamed, J. Ren, H. Lang, and M. El-Gindy, “Optimal Collision Free Path Planning For An Autonomous Articulated Vehicle With Two Trailers,” IEEE international conference industrial technology, pp. 860–865, 2017.

[9] J. w. choi, R. E. Curry and G. H. Elkaim, "Continuous Curvatures Path Generation Based on Bezier Curves for Autonomous Vehicle", International journal of applied mathematics, Vol .40, No.2, pp. 91–102, 2010.

[10] M. Abbas, E. Jamal and J.M. Ali," Bezier Curve Interpolation Constrained By Line", Applied Mathematical Science, Vol.5, No.37, and pp. 1817-1832, 2011.

[11] V. Hlavac, “Manipulator Trajectory Planning”, Czech Technical University in Prague, 2006. Available at http://cmp.felk.cvut.cz/~hlavac.

[12] L. Wang, Y. Liu, and H. Deng, “Obstacle-avoidance Path Planning for Soccer Robots Using Particle Swarm Optimization,” International Conference on Robotics and Biomimetic, no. 1, 2006.