

# Interactive Heuristic D\* Path Planning Solution Based on PSO for Two-Link Robotic Arm in Dynamic Environment

Firas A. Raheem, Umniah I. Hameed

Control and Systems Department, University of Technology, Baghdad, Iraq  
Email: 60124@uotechnology.edu.iq, 61126@uotechnology.edu.iq

**How to cite this paper:** Raheem, F.A. and Hameed, U.I. (2019) Interactive Heuristic D\* Path Planning Solution Based on PSO for Two-Link Robotic Arm in Dynamic Environment. *World Journal of Engineering and Technology*, 7, 80-99.  
<https://doi.org/10.4236/wjet.2019.71005>

**Received:** December 7, 2018

**Accepted:** January 7, 2019

**Published:** January 10, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

This paper is devoted to find an intelligent and safe path for two-link robotic arm in dynamic environment. This paper focuses on computational part of motion planning in completely changing dynamic environment at every motion sample domains, since the local minima and sharp edges are the most common problems in all path planning algorithms. In addition, finding a path solution in a dynamic environment represents a challenge for the robotics researchers, so in this paper, a proposed mixing approach was suggested to overcome all these obstructions. The proposed approach methodology for obtaining robot interactive path planning solution in known dynamic environment utilizes the use of modified heuristic D-star (D\*) algorithm based on the full free Cartesian space analysis at each motion sample with the Particle Swarm Optimization (PSO) technique. Also, a modification on the D\* algorithm has been done to match the dynamic environment requirements by adding stop and return backward cases which is not included in the original D\* algorithm theory. The resultant interactive path solution was computed by taking into consideration the time and position changes of the moving obstacles. Furthermore, to insure the enhancement of the final path length optimality, the PSO technique was used. The simulation results are given to show the effectiveness of the proposed method.

## Keywords

D\* Algorithm, Particle Swarm Optimization (PSO), Path Planning, Two-Link Arm, Known Dynamic Environment

## 1. Introduction

The words “path planning” refer to find a path solution in robotics where the

process is totally or sub automated. It is used to indicate the type of computational process for moving a robot from one place to another with respect to obstacles. In other words, it represents the search for initial feasible path which is considered as the first step to solve the path planning problem and includes finding a path for a robot that must move from initial given start point to the goal point which is also given as the destination position. The environment usually contains a fully de-fined set of obstacles. Therefore, the robot does not collide with any of the obstacles [1]. Generally, obstacles cannot always be static. When the obstacle changes its location over time, it called dynamic. Dynamic environment represents a revolution and important side in modern automation. It redounds in many important diverse applications like air and sea traffic control negotiating, freeway traffic, intelligent vehicles, automated assembly, and automated wheelchairs. Because of existence the moving objects, the robot needs to make decision quickly for avoiding possible collisions [2] [3]. There are three types of dynamic environment. The first type is called a known dynamic environment when all the information regarding the obstacles motions, sizes, shapes, and locations is completely known. The second type is called a partially known dynamic environment when not all information about the obstacles exists at the planning time. In this state, it needs to calculate the robot motion according to insufficient information about the environment. It is important to know when the robot's mission is to complete a map of the environment by scanning the area to make sure if there is a new object found. Finally, when there is nothing to know about the obstacles completely, it's called the totally unknown and dynamic environment and this needs some intelligent methods for sensing and making a proper decision [3].

Due to the rapid increase in the use of robots in different life and industry applications, path planning in dynamic environment is still considered as an actual challenge and needed to be solved interactively for both mobile and manipulator robots. The proposed method in this paper ensures finding an intelligent, proper, safe and relatively optimal path depending on the dynamic environment complexity. The structure of this paper contains seven sections in addition to the introduction, Related Research work, two-link robot arm kinematics, free Cartesian space analysis, proposed method, simulation results, and conclusions.

## 2. Related Research Work

To overcome the dynamic environment challenges, many path planning algorithms have been used in the literature. Here we started with Anthony Stentz [4] whom used The D\* algorithm which is represented the dynamic A\*, it aims to create a short path in real-time by incrementally reforming paths for the robot's state as a new information is found. The D\* extension which concentrate the repairs of significantly reduction the whole time required to subsequent and calculate the initial path re-planning operations which this paper aims to, followed by Yueshi Shen and Knut Huper [5] who suggested using a method to calculate

the optimal joint trajectories for multi-DOF manipulator robot performing constrained motion task. The path planning is completed in two stages, firstly converting the variational calculus problem into a finite dimensional optimization problem, using Newton's method to compute the joint trajectory's inter-mediate points. Secondly, interpolating these points to form the joint curve and adjust it to ensure that the motion constraint will be satisfied throughout the entire trajectory. These calculations are easier to apply for multi-DOF manipulators than the Tow-Link robot arm because multi-DOF manipulators are more flexible.

Many other path planning algorithms were applied for solving dynamic environment problems to find an optimal path for robots in an environment that is only partially known and continuously changing. A method was suggested for generating a collision-free near-optimal path for an autonomous mobile robot in a dynamic environment containing moving and static obstacles using the neural network and fuzzy logic with a genetic algorithm. The neural network has a limited data size therefore, it cannot solve path in bigger environment furthermore, the simulation results show that the method is efficient and gives near-optimal and not optimal path reaching the target position of the mobile robot [6]. An improved virtual force field (VFF) algorithm approach was proposed for the real-time robot path planning, where the environment is unknown and changing. An area ratio parameter with the consideration of the robot size and obstacles was used. Furthermore, a fuzzy control module was added to deal with the problem of obstacle avoidance in dynamic environments. Due to the local minimum problem and the higher computational complexity, the path could not reach optimality [7].

From the same point of view, an algorithm for specifying Cartesian constraints with a single and dual arm operation is proposed based on Rapidly-Exploring Random Tree (RRT). One parameter is used to be the common in all RRT based on planners is the range. The range represents a threshold for the samples drawn newly and their small distance values after that those will add to the tree. Range value will be affective on the planning time which means that it should be selected carefully since the higher values causing short time planning which is advantage but as a result, it also cause a tensive motions and the smaller values give a long time planning and it composed more points in the tree. Also, more checking will be needed for the collision that may occur in the space; therefore it needs extra collision checking [8].

Most of the existing literature in dynamic environment considers only the traditional path planning algorithms and there are only a few studies that consider the heuristic algorithms like Anirudh Vemula [9] whom uses a heuristic based graph search planning algorithm that uses adaptive dimensionality consider time dimension corresponding to region where potential dynamic obstacle collisions can occur. Their results show the approach can only model simple interactions and fail to generalize for complex crowded settings.

Inspired by the above discussion, the contribution of the proposed method in

this paper can be summarized in the following points:

- 1) Multi free Cartesian space analysis has been computed at each motion sample to identify the feasible workspace area.
- 2) Modifying the heuristic D\* algorithm by adding stop case and return backward case for taking into consideration the obstacles position and time which does not used before. These modification cases not included in the original D\* theory.
- 3) Finding a path solution utilizing the proposed method in this paper that use the modified heuristic D\* algorithm based on the PSO optimization technique in known dynamic environment.
- 4) Applying the proposed method to simulate the path solution for the two-link robot arm as case study, which is more difficult than the mass-point case study usually used in path planning researches.
- 5) The resulting path solution from this proposed method can solve the problem of finding an optimal path in dangerous, harsh, toxic environments and inaccessible areas.

### 3. Two-Link Robot Arm Kinematics

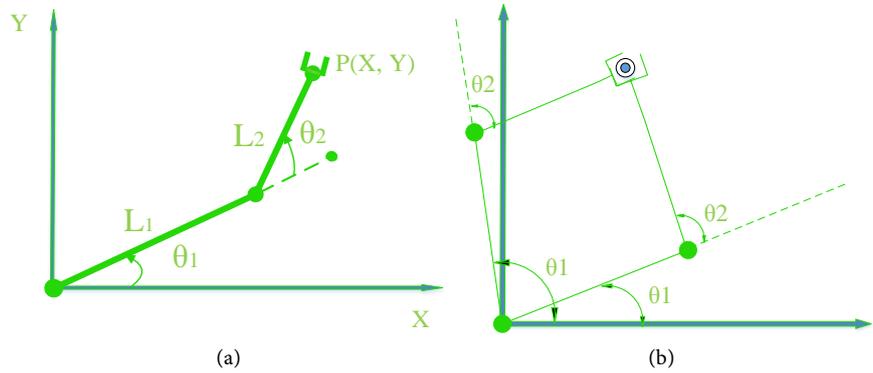
Kinematics is the science of motion that treats the subject without regard to the forces that cause it. Two types of kinematics problems can be classified; the first one is the forward kinematics problem which is concerned with the relationship between the individual joints of the robot manipulator and the position and orientation of the tool or end-effector [10]. The second is the inverse kinematics problem of the serial manipulators has been studied for many decades. It is needed in the control of manipulators. Solving the inverse kinematics is computationally expensive and generally takes a very long time in the real-time control of manipulators. Tasks to be performed by a manipulator are in the Cartesian space, whereas actuators work in joint space. Cartesian space includes orientation matrix and a position vector. However, joint space is represented by joint angles. The conversion of the position and orientation of a manipulator end-effector from Cartesian space to joint space is called an inverse kinematics problem [11]. The planar manipulator is shown in **Figure 1(a)** to illustrate the 2-DOF. The planar manipulators have  $l_1$  and  $l_2$  as their link lengths and  $\theta_1$  and  $\theta_2$  represent joint angles with x and y as task coordinates.

The forward kinematics equations are shown below:

$$x = l_1 \cos \theta_1 \cos \theta_2 + l_2 \cos(\theta_1 + \theta_2) \quad (1)$$

$$y = l_1 \sin \theta_1 \sin \theta_2 + l_2 \sin(\theta_1 + \theta_2) \quad (2)$$

In 2-DOF planner manipulator, the Geometric Solution Approach of inverse kinematics is used for obtaining the joint variables for all the points in Cartesian space regardless their orientation. The calculations of  $\theta_1$  and  $\theta_2$  for the 2-DOF manipulator could be found by applying the following equations.



**Figure 1.** (a) Two degrees of freedom manipulator; (b) The configurations of two-link arm elbow up and elbow down.

Utilizing the cosines law, where the angle is given by:

$$\cos \theta_2 = \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1l_2} = D \tag{3}$$

Since  $\sin \theta_2$  is

$$\sin \theta_2 = \pm\sqrt{1 - D^2} \tag{4}$$

To solve  $\theta_2$ , it has to use the  $\cos^{-1}$  function. Although it is more beneficial to use  $\tan^{-1}$  for numerical solution, in the software implementation, we will use the function  $\tan^{-1}(b/c)$  which is ATAN2 ( $b, c$ ). This function has a uniform accuracy over the range of its arguments, returns a unique value for the angle depending on the signs of  $b$  and  $c$ , and gives the correct solution if  $b$  and/or  $c$  is zero [11].

Subsequently, the two conceivable solutions for  $\theta_2$  can be written as:

$$\theta_2 = \tan^{-1} \frac{\pm\sqrt{1 - D^2}}{D} \tag{5}$$

At last,  $\theta_1$  can be solved by:

$$\theta_1 = \tan^{-1}(y/x) - \tan^{-1} \frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2} \tag{6}$$

where  $l_1$  and  $l_2$  are lengths of the first and the second links, respectively.

#### 4. Free Cartesian Space Analysis

The result of inverse kinematics has been used to build and analyze the workspace in this thesis. The specific points that the end-effector can reach, can be defined as free Cartesian space (FCS). The angles ( $\theta_1$  and  $\theta_2$ ) which were found by inverse kinematics are related to these specific points directly. The shape and volume of the FCS in an environment, which contains many obstacles, are changed depending on the number of obstacles, position and size in addition to the joints limits. In this paper, **Table 1** shows the modeling and simulation. These restrictions impact and limit movement of manipulator and additionally split up the workspace into a reachable region and an inaccessible one.

**Table 1.** Theoretical range of the two-link robotic arm case study.

Link No.	Joint 1	Joint 2
Ranges	$0 \leq \theta_1 \leq 360$	$-90 \leq \theta_2 \leq 90$

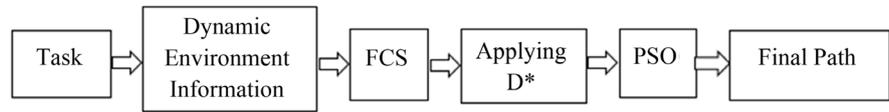
To find all possible solutions, we should compute the FCS and that done was by analyzing the environment's points. The center point and the diameter of the obstacles are the basic issues upon which the checking function depends. Moreover, there are three possible states for each point in the whole environment. Firstly and as shown in **Figure 1(b)**, there are two solutions to each point, the elbow up solution and the elbow down solution. But in the second state, there is just one solution, either elbow up or elbow down, and in the third one, there is no solution because the point lies out of the reachable area of the manipulator or on an obstacle, or the one or two links collide with an obstacle [12].

The resulted areas are separated into three parts, the first one contain the points which can be reached by the elbow up solution, and the second area contains the points which can be reached by the elbow down solution, and the third one represents the unreachable points that contain points outside the accessible range of manipulator workspace, points in obstacle's body and points that make collision with the obstacles. The biggest factor that affects the free Cartesian is the length of the arm link [12].

## 5. Proposed Method: D\* Algorithm Based on PSO

After applying the D\* algorithm to find the non-interactive path solution in known dynamic environment and studying the produced path, it has been proposed to make the environment interactive considering the position and time of the obstacles to enhance the path planning. Employing the D\* algorithm in the known obstacle position and time dynamic environment, to find the robot path needs to build a map with its limitation, start and target locations and the obstacles locations and dimensions. The proposed planning for the two-link robot arm in dynamic environment is shown in **Figure 2**. In the interactive path solution of known dynamic environment, in each move of the robot from node to node in each motion sample the free Cartesian space (FCS) will be calculated. As a result, the movement range of the robot will be defined and described accurately. It is a more intelligent behavior having multiple FCS with each motion sample the obstacles move. The new FCS, which was calculated, gives an indication to find the feasible area for the robot to move on. This process takes time but gives an accurate decision for the environment and high accuracy to the robot movement in the environment.

Since the process is off-line and after constructing the environment, the algorithm will start its work when the robot firstly stands on the target node and starts the initial expand to its 8-connected neighbor. Each node will have an initial cost function where the robot will choose the least cost function node to move on and put it in the closed list. The other nodes will go to the open list



**Figure 2.** Proposed planning for the two-Link robotic arm in dynamic environment.

with taking into consideration the obstacles having a huge cost function and sorting the initial locations in the closed list. With each motion samples, the closed list will be updated to catch the new obstacle location. This means, the closed list is changing dynamically over motion samples to save the obstacle nodes in the current motion sample only, and the process complete like that until reaching the start node. When the search is completed the robot will move from the start node with respect to the closed list instantaneously to have information about free node for moving on towards the target.

$$f = g + h \tag{7}$$

where  $f$  is objective function and  $g$  is the estimated cost from start point while  $h$  is the cost to the goal.

The velocity of the moving obstacle is calculated by:

$$v_x = \frac{\Delta x}{MS \times T_s}, \quad v_y = \frac{\Delta y}{MS \times T_s}$$

where  $\Delta x$  represents the change of distance in X-axis,  $\Delta y$  represents the change of distance in Y-axis and  $T_s$  represents the sample time which assumed to be 0.01 sec.

After finding the  $D^*$  path, the PSO optimization technique is used for enhancing the path, removing sharp edges and making it the shortest because the  $D^*$  path has stairs shape sometimes. The principle of PSO technique depends on generated specified number of particles in random positions in the specified workspace, also the velocity of those particles is nominated randomly, each particle has a memory that stores all the best position have been visited before, in addition to the fitness in that position which has been improved over time [13].

In each iteration of PSO technique, the  $pos_{ij}$  and  $vel_{i,j}^t$  vectors of particle  $i$  is modified in each dimension  $j$  in order to lead the particle  $i$  toward either the personal best vector ( $pbest_{ij}$ ) or the swarm's best vector ( $gbest_{ij}$ )

$$vel_{i,j}^{t+1} = w \times vel_{i,j}^t + c_1 \times r_{1,j} \times (pbest_{ij} - pos_{ij}) + c_2 \times r_{2,j} \times (gbest_{ij} - pos_{ij}) \tag{8}$$

The position of each particle (the point that is chosen from  $D^*$  path) is updated by using the new velocity for that particle, according to equation below:

$$pos_{ij}^{t+1} = pos_{ij}^t + vel_{ij}^{t+1} \tag{9}$$

where  $c_1$  and  $c_2$  are the cognitive coefficients ( $c_1 + c_2 \leq 4$ ), and  $r_{1,j}$  and  $r_{2,j}$  are random real numbers between [0, 1], the inertia weight  $w$  controls the particle momentum.

The value of  $vel_{ij}$  is clamped to the range  $[-vel_{max}, vel_{max}]$  to reduce the probability of leaving the search space by the particle. If the search space is de-

finned by the bounds  $[-pos_{max}, pos_{max}]$ , then the value of  $vel_{max}$  is typically set so that  $vel_{max} = k * pos_{max}$ , where  $0.1 \leq k \leq 1.0$ .

A large inertia weight ( $w$ ) eases the local when its value is large and eases the global search when its value is small [14].

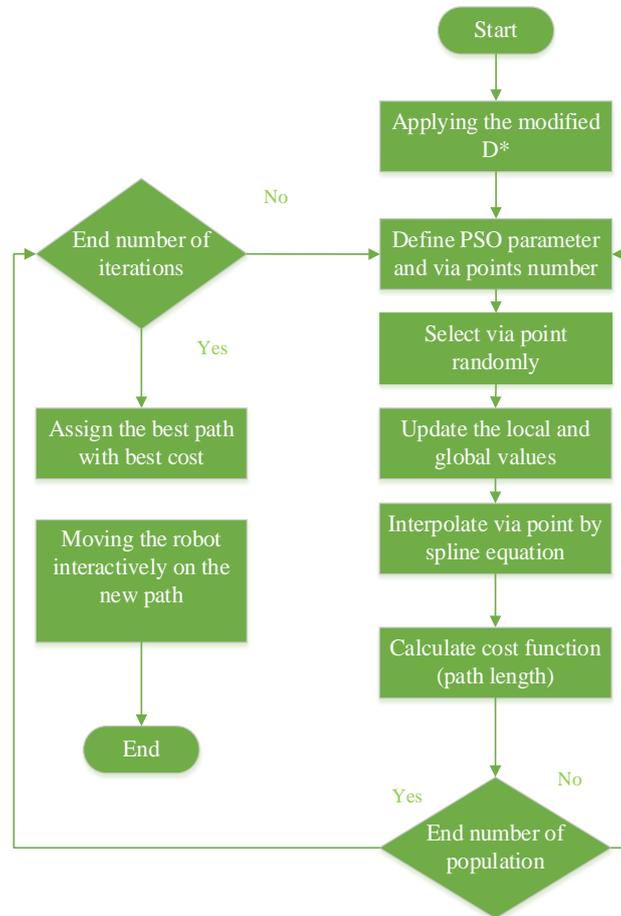
The first step to operate the PSO is to set the parameters of PSO ( $c_1 = 1$ ,  $c_2 = 3$ ,  $w = 1$ ) which are found by a trial and error technique to reach the best results. Now, the D\* path enters the PSO work, which will represent the D\* path as the search space and choose the optimal points from it taking into account the general shape for the D\* path and without distorting the original form. After determining the PSO search space which is the D\* path matrix, the search process begins with defining the number of via points (the points that is chosen from D\* path) that search for the new points according to the search space dimension and the PSO parameters. These via points are chosen randomly from the search area (D\* path) and will represent the position parameters in the PSO equation.

The cubic spline interpolation equation has been used to connect the best chosen points from the PSO optimization technique to generate the corresponding smoothed path. The cost function of PSO, which is the path length, is computed. Then the elementary velocity is set, and updating the local and global parameters are the next step. This procedure is reiterated until the given points that are looked over the path are finished. The next step is updating the velocity and position equations, for converging the via points into the best cost, and also updating the local and global parameters. The procedure is repeated for the number of specified populations and the entire process is repeated for a number of specified iterations. The global values that construct the shortest path are then allocated. The entire process flowchart is represented in **Figure 3**. After getting the smoothed path, the robot will move also interactively with position and time on the new PSO path.

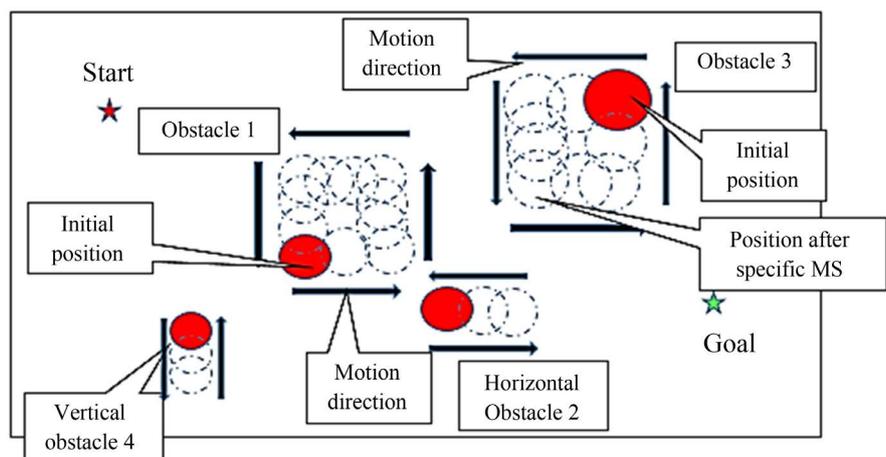
## 6. Simulation Result

### 6.1. First Environment

Here we have applied the two-Link robot arm as a case study. The arm length is equal to the half of the environment with 50 cm for each link and the arm has 360 degrees. The obstacle motion behavior is shown in **Figure 4**. Since the multi FCS is applied as clarified in **Figure 5**, where the column (5(a)) represents the elbow down FCS analysis at some selected motion samples; 1, 4, 5, 7, and 8, while in the column (5(b)) and (5(c)) represents the elbow up FCS analysis and the gathering between elbow up and elbow down FCS analysis respectively at the same motion samples. In each FCS figure the colored area (blue, red or green) is represented the free and safe area for moving and the white area is represented the unreachable area. Therefore, it has calculated the elbow up and elbow down FCS to enable the robot from making decision which free space is the suitable for moving. After computing these calculations the robot arm could move more freely because it could detect with each FCS the free area to move on. **Figure 6**

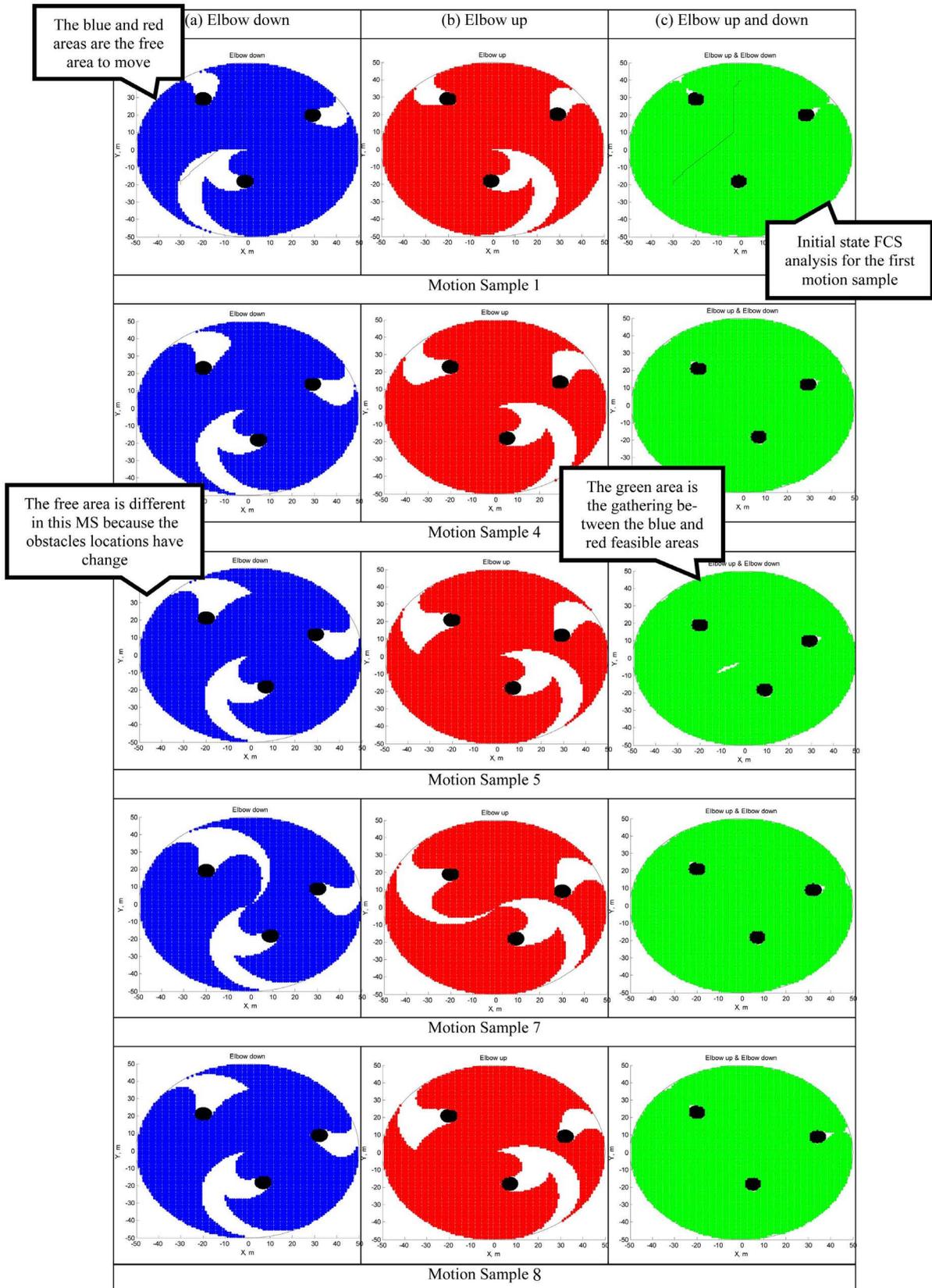


**Figure 3.** Proposed D\* via Point flowchart for position and time consideration dynamic environment.



**Figure 4.** Obstacle motion behavior.

shows the two-link robot arm D\* path environment where part (a) represents the beginning of the planning while the end-effector is on the start point, part (b) is the arm location after 8 motion sample from planning process. At part (c) the arm is return backward at motion sample 15 to avoid collision, part (d) and



**Figure 5.** FCS for the D\* path and PSO path where (a) represents elbow down, (b) represents elbow up and (c) represents gathering between elbow up and elbow down.

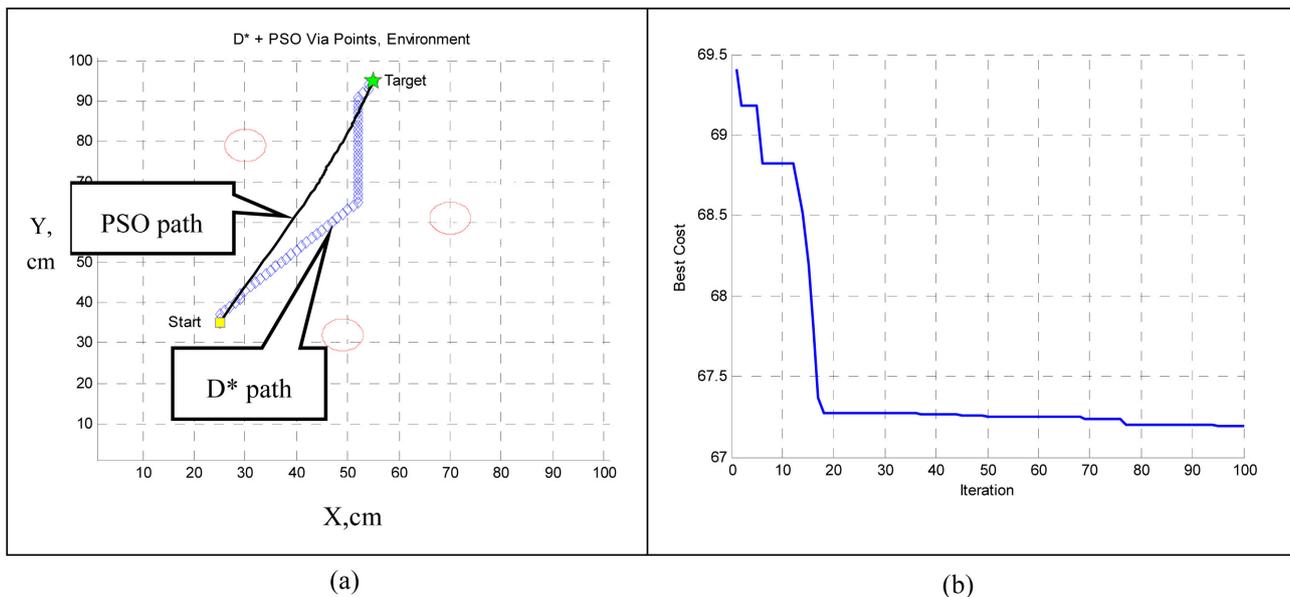


(e) showed the planning after 21 and 36 motion sample respectively and after 38 motion sample the arm is returned backward to evade the obstacle then it reached the goal point safely as shown in part (g) and (h). Where the process is off-line, the PSO optimization technique is used to get rid of sharp edge and shorten the path as shown in **Figure 7(a)**, where the blue line represents the D\* path and the black line represents the PSO path. Also, **Figure 7(b)** clarified that the D\* path based on PSO decreased by 3.2846 cm long from the original path. **Figure 8** shows the D\* based on PSO path planning process from the initial state in part (a) where the end-effector is on the start location to the part (f) where the process has completed by reaching goal location. At part (b) where the motion sample equal 7 the arm has restarted backward to safe itself from collision. In addition, **Figure 9** and **Figure 10** represent the two link robot arm joint angles and change in these angles for the optimized path respectively. The same procedure will be done for the second environment.

These environments featured the interactive style motion where the robot should make a decision and must be stopped in some situations if it is needed, in this case, the stop state and the backward motion guarantee a collision free path. **Table 2** clarifies the environment information.

### 6.2. Second Environment

For more verification, a second more complicated environment with four moving obstacles in such different way in comparing with the first environment and more close to the robot links. Two from these obstacles move in a square motion behavior as in **Figure 4**. Since the multi FCS is applied as clarified in **Figure 11**, where the column (11(a)) represents the elbow down FCS analysis at some selected motion samples; 1, 5, 7, 8, and 11, while in the columns (11(b)) and



**Figure 7.** (a) Final D\* and PSO path and (b) the path cost function.

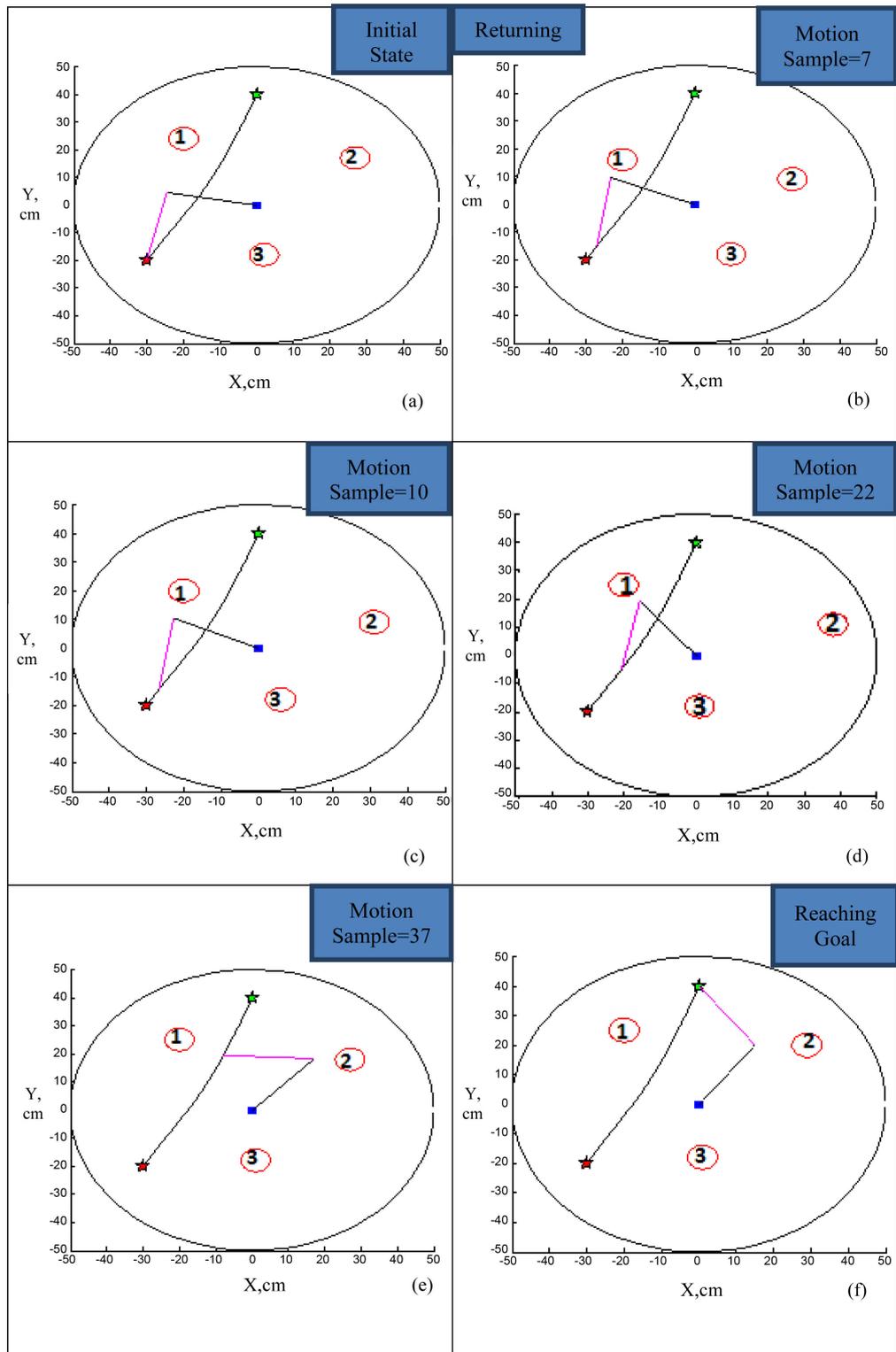
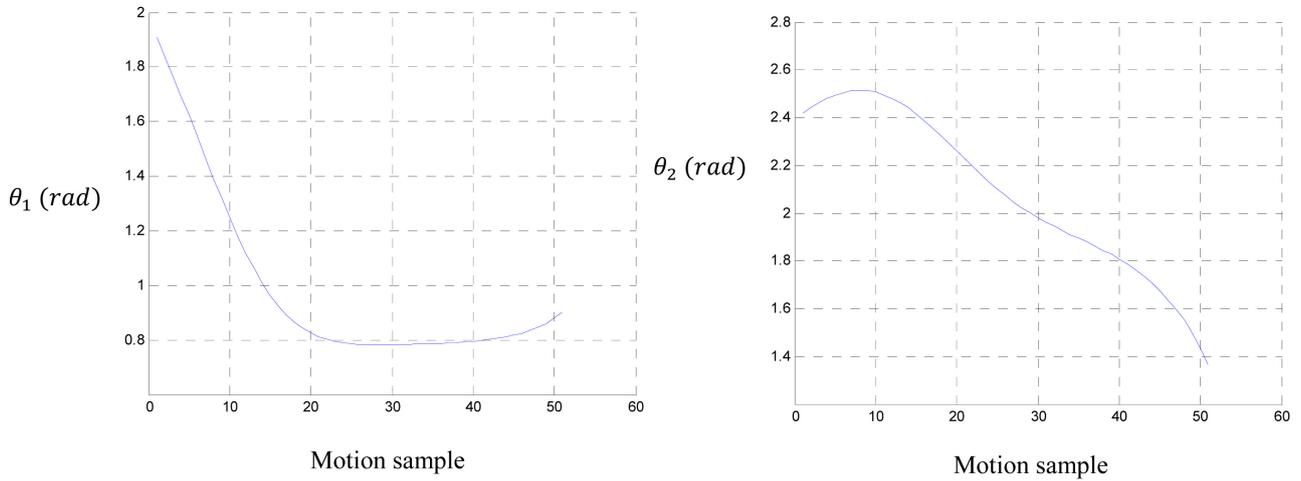
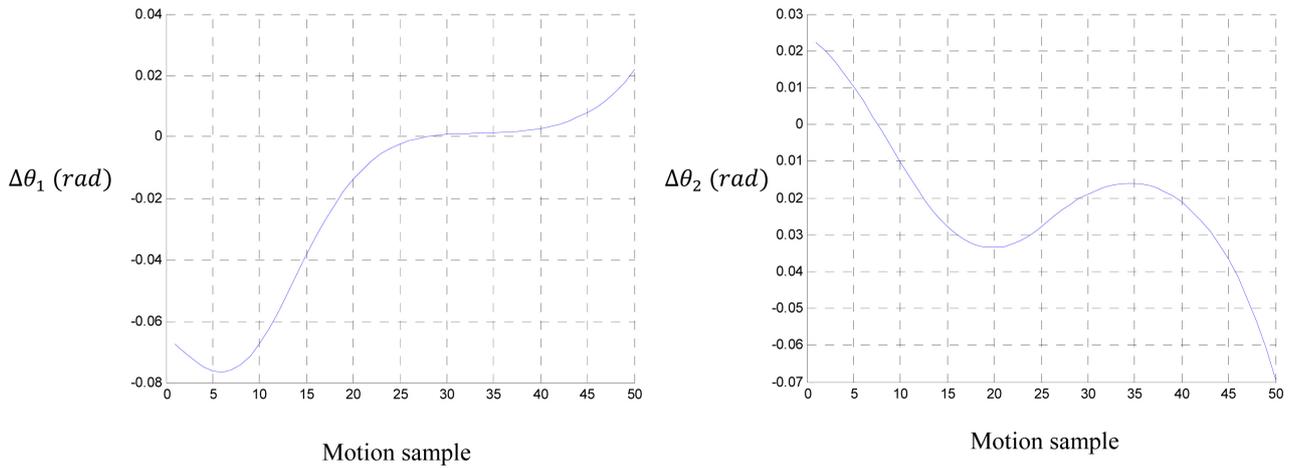


Figure 8. Two-link robot arm result for the D\* based on PSO path.

(11(c)) represents the elbow up FCS analysis and the gathering between elbow up and elbow down FCS analysis respectively at the same motion samples. In each FCS figures the colored area (blue, red or green) indicates the free and safe



**Figure 9.** The two-link robot arm joint angles for the optimized path.

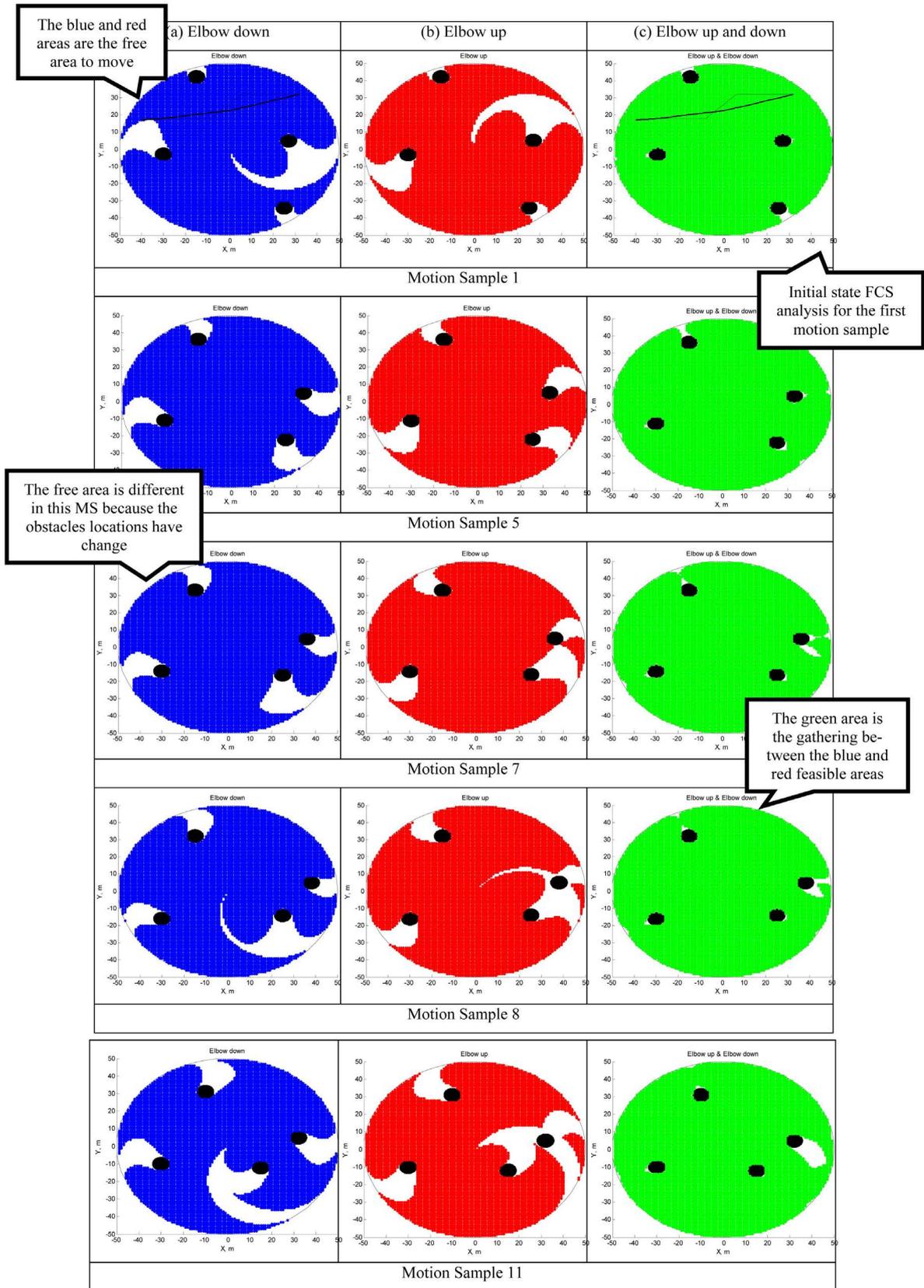


**Figure 10.** The change in joint angles for the optimized path.

**Table 2.** Environments information.

Environment	1	2
Obstacles number	3	4
Population	50	50
iteration	100	100
D* path	72.4264 cm	78.2132 cm
PSO path	69.6865 cm	73.8408 cm

area for moving robot links while the white area indicates the unreachable area. Therefore, the robot can take a proper decision for selecting the next points in a free and intelligent path inside the permitted motion area. **Figure 12** shows the two-link robot arm D\* path environment where part (a) represents the beginning of the planning while the end-effector is on the start point, part (b) shows the arm location after 11 motion samples, part (c) indicates that the arm is returning backward case at the motion sample 21 to avoid collision, part (d) and



**Figure 11.** FCS for the D\* path and PSO path for the second environment where (a) represents elbow down, (b) represents elbow up and (c) represents gathering between elbow up and elbow down.

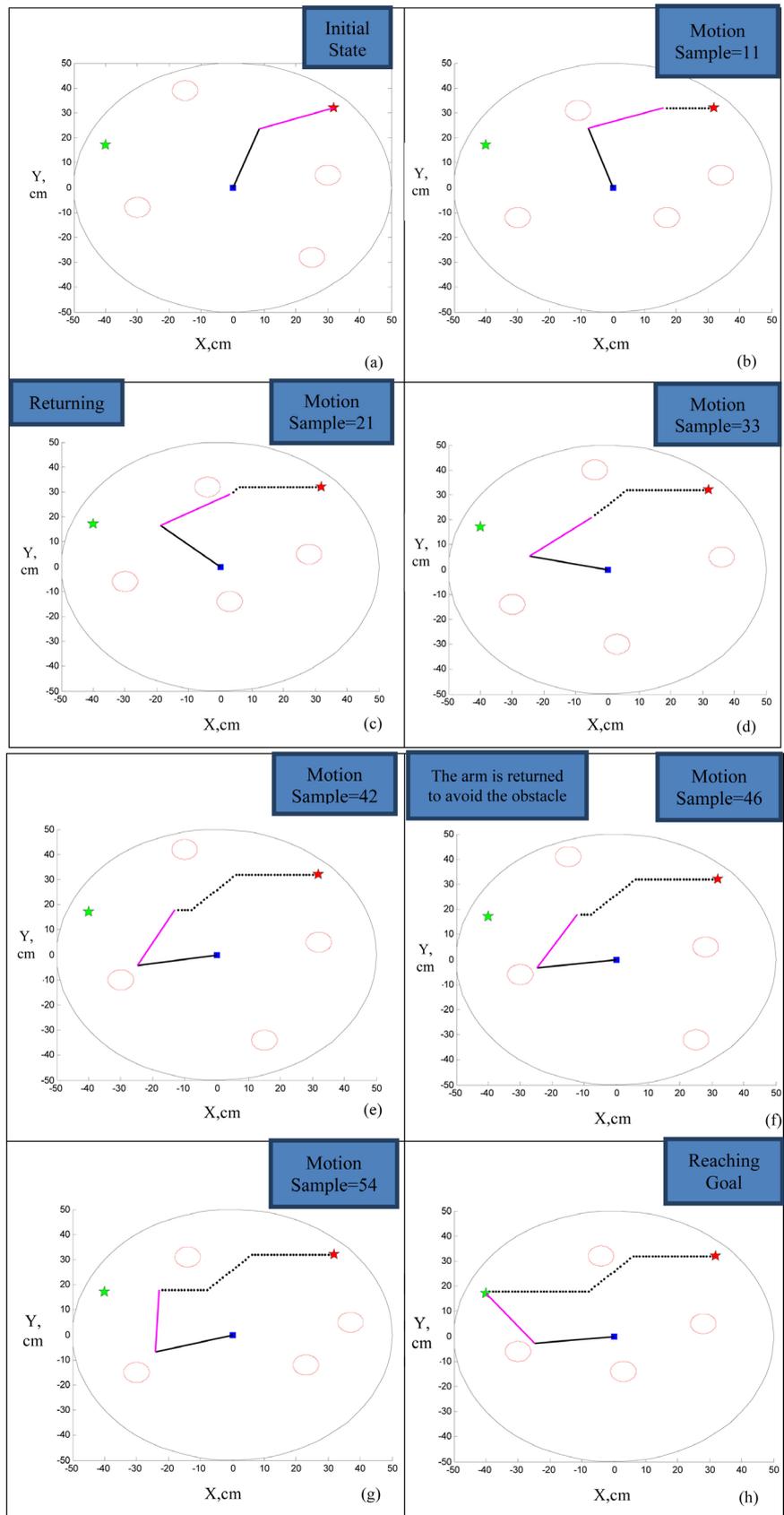


Figure 12. The two-link robot arm result for the D\* path for the second environment.

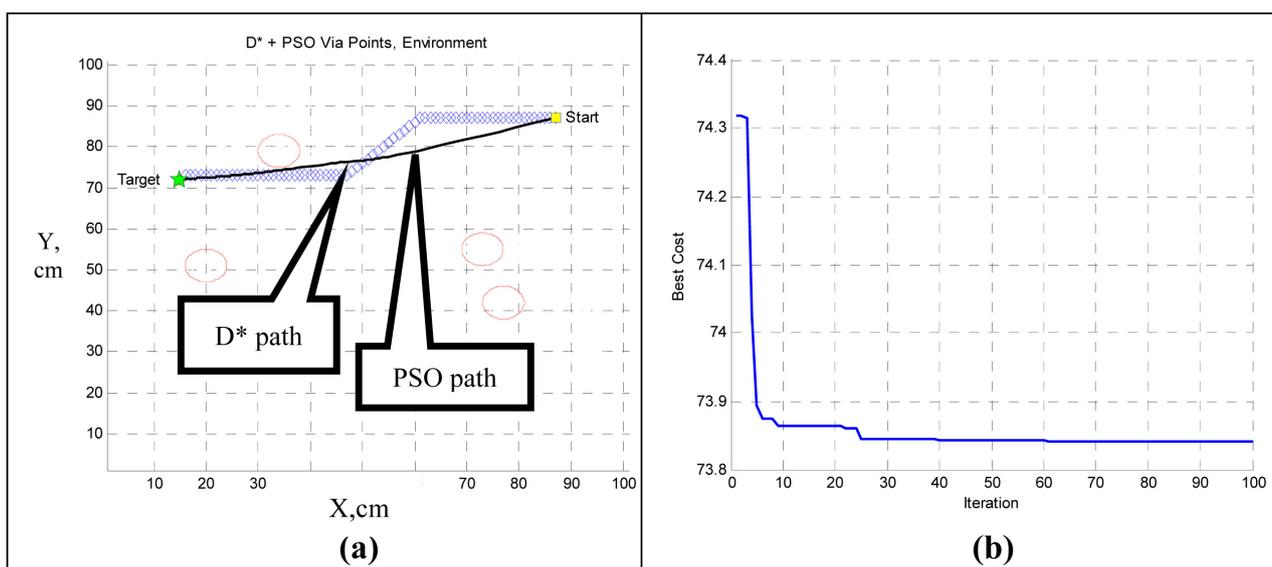
part (e) show the planning after 33 and 42 motion samples respectively, while after 46 motion samples, the arm is returned backward again to evade the closest obstacle before reaching the goal point safely as shown in part (g) and (h).

Because of the known position and time motion behavior of the obstacles, the process of the path planning can be made off-line, therefore the PSO optimization technique can play an important role for finalizing this process and finding the best possible shortest and optimal path to get rid of the heuristic sharp path edges. Both of the D\* and PSO paths are shown in **Figure 13(a)**, where the blue line represents the D\* path and the black line represents the PSO path. Also, **Figure 13(b)** represents the optimization cost function progress and proves that the final D\* based on PSO path length was shorter by 4.3724 cm than the original path. **Figure 14** confirms the path safety validation by showing clearly all the robot configurations during motion according to the optimized D\* based on PSO path from the initial point as in part (a) to the goal point as in part (f). In addition, **Figure 15** and **Figure 16** represent the two link robot arm joint angles and the changes in these angles during the optimized path respectively. Also, **Table 2** clarifies the environment information.

In comparison with the nearest research literature [4] that try to solve the same problem for mobile robot using the D\* heuristic method, our proposed method has different results because of our original modification that applied on D\* algorithm which guarantees the interactivity and safety of the path and because of finalizing the path using PSO optimization technique.

### 7. Conclusion

This paper proposed an optimized heuristic approach for solving the problem of two-link robot arm path planning in known position and time changes dynamic environment. This approach utilizes the use of heuristic D\* algorithm for finding



**Figure 13.** (a) Final D\* and PSO path and (b) the path cost function for the second environment.

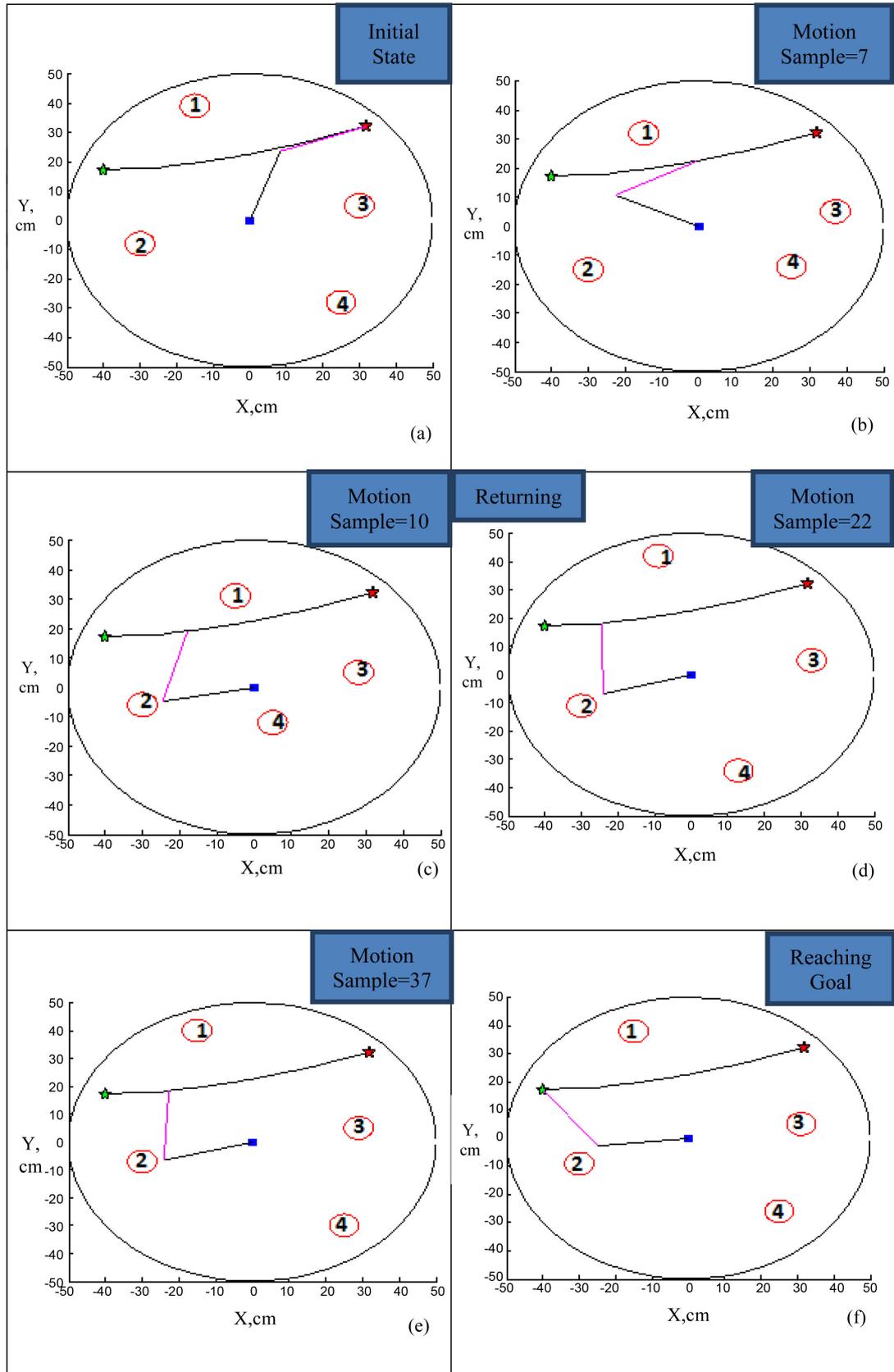
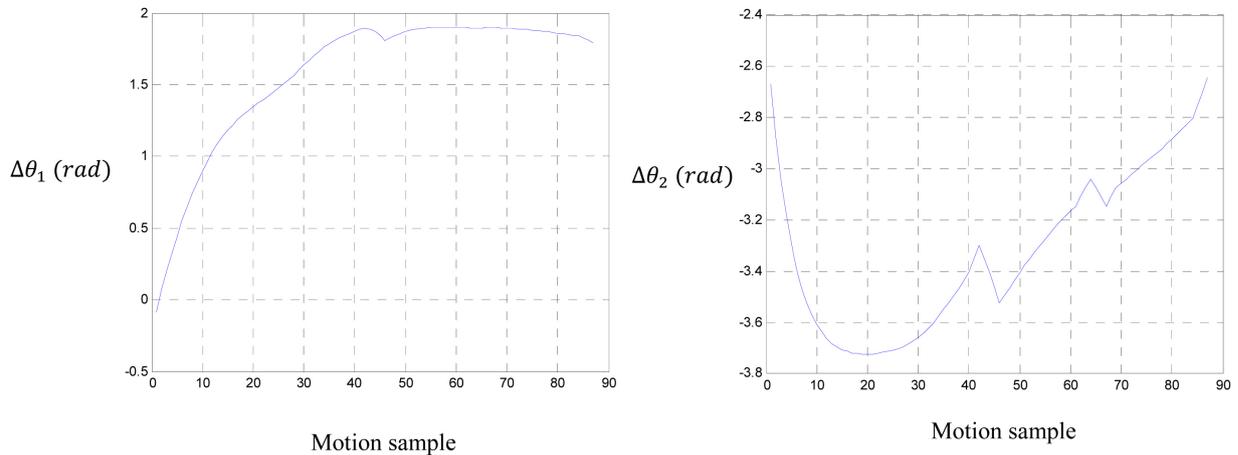
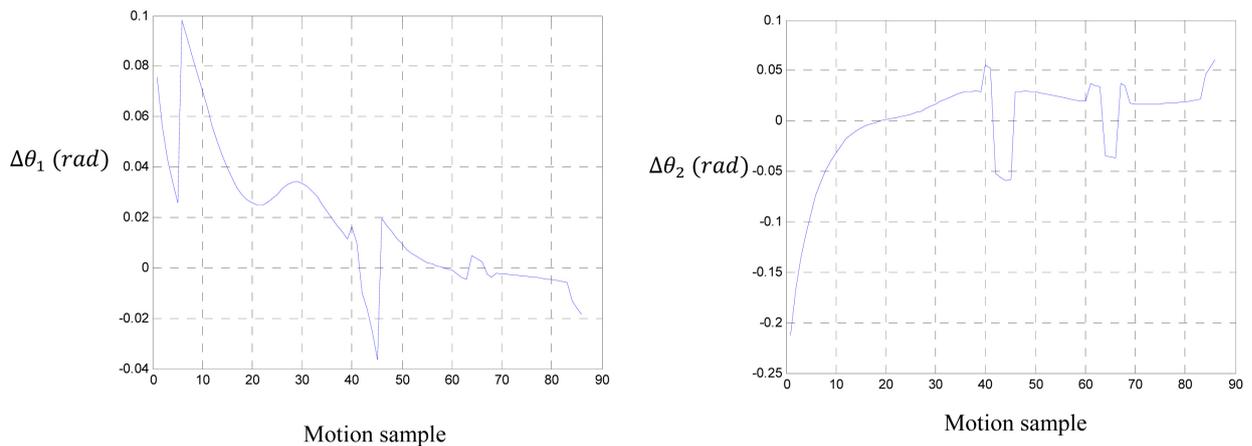


Figure 14. The two-link robot arm result for the D\* based on PSO path for the second environment.



**Figure 15.** The two-link robot arm joint angles for the optimized path of the second environment.



**Figure 16.** The change in joint angles for the optimized path of the second environment

the best path and the use of PSO optimization technique for getting the final optimal path. Also, a modification on the D\* algorithm has been done by adding a stop case and return backward case to ensure the path solution interactivity response to the obstacles position and time continuous changes during robot motion. From this approach, it was possible to get an intelligent path in spite of the environmental difficulties where a completely collision free path could be found at the expense of optimality. Therefore, the PSO optimization technique has been used to enhance the final path by removing the sharp edges until reaching path optimality. The result clearly proves the advantage of using this proposed methodology since the robotic arm has preserved the intelligence interactive behavior even after applying PSO Technique. The length of the final path was smart, smooth, safe, and optimal, and traditional D\* algorithm could not find such a path. Our proposed method has a new approach of solution that leads to different results since it keeps the motion behavior of the robot to be intelligent and interactive with dynamic environment changes. The future directions for developing this proposed method include utilizing other intelligent methods and theories.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Miao, H., Tian, Y.-C. and Feng, Y.M. (2013) Robot Path Planning in Dynamic Environments using a Simulated Annealing Based Approach. Faculty of Science and Technology, Queensland University of Technology, Brisbane.
- [2] Raheem, F.A. and Bader, M.M. (2017) Development of Modified Path Planning Algorithm Using Artificial Potential Field (APF) Based on PSO for Factors Optimization. *American Scientific Research Journal for Engineering, Technology, and Science*, **37**, 316-328.
- [3] Fujimura, K. (1991) Motion Planning in Dynamic Environments. Springer-Verlag, Tokyo. <https://doi.org/10.1007/978-4-431-68165-6>
- [4] Stentz, A. (1995) The Focused D\* Algorithm for Real-Time Replanning. *Proceedings of the International Joint Conference on Artificial Intelligence*, Montreal, 20-25 August 1995, 1652-1659.
- [5] Shen, Y.S. and Huper, K. (2004) Optimal Joint Trajectory Planning for Manipulator Robot Performing Constrained Motion Tasks. Department of Information Engineering, Australian National University, National ICT Australia Limited, Canberra.
- [6] Khelchandra, T. and Huang, J. (2015) Robot Motion Planning with Neuro-Genetic-Fuzzy Approach in Dynamic Environment. *International Journal of Information Technology*, **21**, 1-23.
- [7] Ni, J.J., Wu, W.B., Shen, J.R. and Fan, X.N. (2014) An Improved VFF Approach for Robot Path Planning in Unknown and Dynamic Environments. *Mathematical Problems in Engineering*, **2014**, Article ID: 461237. <https://doi.org/10.1155/2014/461237>
- [8] Asadi, B. (2015) Single and Dual Arm Manipulator Motion Planning Library. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 28 September-2 October 2015.
- [9] Vemula, A. (2017) Safe and Efficient Navigation in Dynamic Environments. Carnegie Mellon University, Pittsburgh.
- [10] Spong, M.W., Hutchinson, S. and Vidyasagar, M. (2005) Robot Modeling and Control. 1st Edition, Wiley, Hoboken.
- [11] Craig, J.J. (2005) Introduction to Robotics Mechanics and Control. 3rd Edition, Pearson Prentice Hall Pearson Education, Inc., Upper Saddle River, NJ.
- [12] Sadiq, A.T., Raheem, F.A. and Abbas, N.A.F. (2017) Optimal Trajectory Planning of 2-DOF Robot Arm Using the Integration of PSO Based on D\* Algorithm and Cubic Polynomial Equation. *The First International Conference for Engineering Researches*, Baghdad, March 2017, 458-467.
- [13] Boonyaritdachochoi, P., et al. (2010) Optimal Congestion Management in an Electricity Market Using Particle Swarm Optimization with Time-Varying Acceleration Coefficients. *Computers & Mathematics with Applications*, in press.
- [14] Poli, R., Kennedy, J. and Blackwell, T. (2007) Particle Swarm Optimization: An Overview. *Swarm Intelligence*, **1**, 33-57.