Al-Khwarizmi Engineering Journal, Vol. 15, No. 2, June , (2019)
P.P. 108- 123

Al-Khwarizmi Engineering Journal

Heuristic D* Algorithm Based on Particle Swarm Optimization for Path Planning of Two-Link Robot Arm in Dynamic Environment

Firas A. Raheem *　　**Umniah I. Hameed****

**, **Department of Control and Systems Engineering/ University of Technology*

*Email: 60124@uotechnology.edu.iq

**Email: 61126@uotechnology.edu.iq

(Received 29 November 2018; accepted 14 January 2019)

https://doi.org/10.22153/kej.2019.01.004

Abstract

Finding a path solution in a dynamic environment represents a challenge for the robotics researchers, furthermore, it is the main issue for autonomous robots and manipulators since nowadays the world is looking forward to this challenge. The collision free path for robot in an environment with moving obstacles such as different objects, humans, animals or other robots is considered as an actual problem that needs to be solved. In addition, the local minima and sharp edges are the most common problems in all path planning algorithms. The main objective of this work is to overcome these problems by demonstrating the robot path planning and obstacle avoidance using D star (D*) algorithm based on Particle Swarm Optimization (PSO) technique. Moreover, this work focuses on computational part of motion planning in completely changing dynamic environment at every motion sample domains. Since the environment type that discussed here is a known dynamic environment, the solution approach can be off-line. The main advantage of the off-line planning is that a global optimal path solution is always obtained, which is able to overcome all the difficulties caused by the dynamic behavior of the obstacles. A mixing approach of robot path planning using the heuristic method D* algorithm based on optimization technique is used. The heuristic D* method is chosen for finding the shortest path. Furthermore, to insure the path length optimality and for enhancing the final path, PSO technique has been utilized. The robot type has been used here is the two-link robot arm which represents a more difficult case than the mobile robot. Simulation results are given to show the effectiveness of the proposed method which clearly shows a completely safe and short path.

Keywords: *D* Algorithm, known Dynamic Environment, Particle Swarm Optimization, Path Planning, Two-Link Arm.*

1. Introduction

The words “path planning” in robotics where the process is totally or sub automated. It is used to refer to this type of computational process for moving an object from one place to another with respect to obstacles. In other word it is searching for initial feasible path and this is the first step in the planning process for the robot path. The path planning problem is that of finding a path for a robot that must move from initial point, which is given as the start position to the goal point which is given as the destination position, in an environment that contains a fully defined set of obstacles. Therefore, the robot does not collide with any of the obstacles [1].

Generally, obstacles cannot always be static. When the object stays motionless in the environment, it calls static, i.e., fixed constantly at a location, but when the object changes its location over time, it calls dynamic. Dynamic environment represents a revolution and important side in modern automation, it redound in many important diverse applications like, air and sea traffic control negotiating, freeway traffic, intelligent vehicles, automated assembly and automated wheel chairs. Because of existence the moving objects, the robot needs to make decision quickly for avoiding possibility collisions. This represents the mutual part among these applications. The ability to deal with moving objects is needful for the navigation of any factual robot. The range of applications of intelligent robots could be widely increased since it's able to handle moving objects [2-3].

There are three types of dynamic environment, where all information is regarding to the obstacles, i.e., motions, sizes, shapes, locations, etc., is completely known, the obstacle is said to be in a known dynamic environment. As example for known environment the automated production line. Partially known Dynamic Environment is the second type when not the all information about the obstacle are exist at the planning time. In this state, it needs to calculate the robot motion with 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 or to scan a certain area to make sure if there is a new object is found in the area. Finally when there is nothing to know about the obstacle it's called the totally unknown dynamic environment and this need intelligent ways to be used. As example for unknown and partially known environment the other robots and humans in workspace [3]. Path planning in dynamic environment is considered as an essential problem for manipulators and autonomous robots therefore, the fundamental issue of this work is to show the robot path planning in a dynamic environment and moving obstacle avoidance utilizing D* algorithm to fulfill the accompanying necessities.

2. Review of Previous Researches

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 Enric Cervera I Mateu [5] who suggested using robot programming architecture to learn fine motion tasks such as autonomous process of experience repetition could carry out the goal in minimized steps. This involves uncertainty in location, and the robot will be guided with sensory information by using force sensor but this work involves uncertainty in location.

Many other path planning algorithms were applied for solving dynamic environment problems such as deriving the non-linear v-obstacle for general planar obstacles to be useful in analytic expressions. Considering obstacles moving on arbitrarily trajectories is represents the concept of counting velocity obstacles. The nonlinear v-obstacle has been consisted of a deformed cone i.e. a time-scaled map of the obstacles that extend along their trajectories. Avoiding the obstacle during the time interval for which the v-obstacle has been generated could be achieved by selecting a single velocity vector outside the nonlinear v-obstacle. This method works accurate only with mobile robot [6]. As well as this, finding out a reasonable relation among parameters that used in the path planning strategies in a platform, where a robot will be able to move freely from the start location in a dynamic environment with map and make an optimal path to a set target location without collision with the static and moving objects. Researchers used a new optimal type algorithm of Rapidly-exploring Random Tree RRT algorithm, named RRT*. The mechanism of this strategy is based on incremental sampling that covers the all space and will act quickly. It can be used in more than single dimension environments and will be computationally efficient but they have found that a high number of sampling doesn't give the best solution in addition to the higher computational time that was required for finding the path [7].

From a point of view, algorithm for specify Cartesian constraints in a single and dual arm operation is proposed and became a library for solving manipulator motion planning problems that has been developed. The main algorithm is based on RRT. One parameter is used to be the common in all RRT based on planers 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 be 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. This reduction in search space supply the approach in great speed-ups in planning time over traditional approaches such as heuristic-based A*, will act without losing safety, dynamic feasibility and bounded sub-optimality of the solution This 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:

- Finding path solution utilizing the heuristic D* algorithm based on the PSO optimization technique in known dynamic environment.
- Applying the proposed method using two-link robot arm, which does not used before.
- This work 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) to illustrate the 2-DOF. The planar manipulators have l_1 and l_2 as their link lengths and θ_1 and θ_2 represent joint angles with x and y as task coordinates.

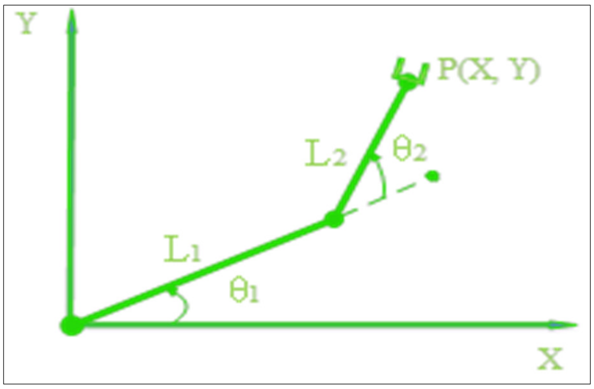


Fig. 1. Two degrees of freedom manipulator.

The forward kinematics equations are shown below:

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

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

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 θ_1 and θ_2 for the 2-DOF manipulator could be found by applying the following equations.

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

$$\cos\theta_2 = \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2 l_1 l_2} = D \qquad \ldots (3)$$

Since $\sin\theta_2$ is

$$\sin\theta_2 = \mp\sqrt{1 - D^2} \qquad \ldots (4)$$

To solve θ_2 , it has to use the cos^{-1}function. Although it is more benefical to use $tan^{-1}$for numerical solution [11].

Subsequently, the two conceivable solutions for θ_2 can be written as:

$$\theta_2 = \tan^{-1}\frac{\mp\sqrt{1-D^2}}{D} \qquad \ldots (5)$$

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

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

Where $l_1$and l_2 are lengths of 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 work. The specific points that the end-effector can reach, can be defined as free Cartesian space (FCS). The angles (θ_1 and θ_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 order to model and simulate the motion of two-link robot arm according to plan the path, the theoretical range of the two-link arm shown in table (1). These restrictions impact and limit movement of manipulator and additionally split up the workspace into a reachable region and an inaccessible one.

Table 1,
The theoretical range of the two-link Robot Arm

Link No.	**Ranges**
Joint 1	$0 \le \theta_1 \le 360$
joint 2	$-90 \le \theta_2 \le 90$

To find all possible solutions, should be computed the FCS and that was done by analyzing the environment's points. The center point and the diameter of the obstacles are the basic issues upon which the checking function depend. Moreover, there are four possible states for each point in the whole environment. The first state is the elbow up solution, the second state is the elbow down solution and the third state is the elbow up and down solution where the arm could reach the target either in elbow up solution or elbow down solution and as shown in figure (2). In addition, there is the last state the no solution case 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].

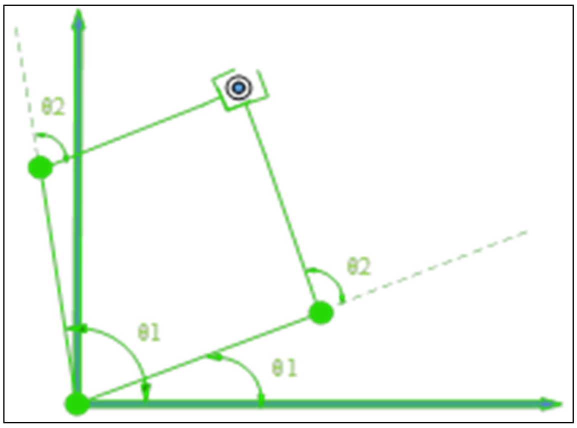


Fig. 2. The configurations of two-link arm elbow up and elbow down.

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. Heuristic D* Algorithm

The D* algorithm is an algorithm that is generate short path in real-time by a graph with update or change arc costs. The utility of D* is that the most of arc cost corrections occurs in the robot's neighborhood. From this corrections, the only part of the path is going to out to the robot's current location is need for replanting. The partial optimal cost map are keep limited for the locations that the robot may use. Furthermore, the states that is probable to produce a new optimal path is used to reform of the map's cost which will be limited and re-entrant partially. D* sets the conditions for determining when repair will be stood, because of finding a new path or the previous path has been found is still optimal. So, the D* strategy is used to say that it is very memory active, works in unbounded environments and computationally as well. The mechanism of the D* algorithm is designed repeat finding optimal path to the directed graph, which means that the arc costs are assorted with transference cost values that could be ranged over a continuation. The corrections of the arc cost that came from the sensors it could be made in any time, also the estimated and measured arc costs that known is comprised to the environment map. This strategy could be used in any planning representation, including grid cell structures and visibility graphs. [13]

$$f = g + h \qquad \ldots (7)$$

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

6. Particle Swarm Optimization Technique Theory (PSO)

PSO is a laborious stochastic optimization technique, it depends on the intelligence and movement of birds swarms. Inspired by the behavior of nature social and dynamic motions with communications of fish schooling, insects or birds flocking. A number of basic variations have been developed due to improve speed of convergence and quality of solution found by the PSO. 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 [14-15].

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 ($gbest_{ij}$)

$$\mathrm{vel}_{i,j}^{t+1} = \mathrm{w} \times \mathrm{vel}_{i,j}^{t} + \mathrm{c}_1 \times \mathrm{r}_{1,j} \times \left(\mathrm{pbest}_{ij} - \mathrm{pos}_{ij}\right) + \mathrm{c}_2 \times \mathrm{r}_{2,j} \times \left(\mathrm{gbest}_{ij} - \mathrm{pos}_{ij}\right) \quad \ldots (8)$$

The position of each particle (bird) is updated by using the new velocity for that particle, according to equation below:

$$\mathrm{pos}_{ij}^{t+1} = \mathrm{pos}_{ij}^{t} + \mathrm{vel}_{ij}^{t+1} \quad \ldots (9)$$

Where c_1 and c_2 are the cognitive coefficients (c_1+c_2<=4), and $r_{1,j}$ and $r_{2,j}$ are random real numbers between [0, 1], the inertia weight w controls the particle momentum [16-17].

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 defined by the bounds [-pos_{max}, pos_{max}], then the value of vel_{max} is typically set so that $vel_{max} = k * pos_{max}$, where $0.1 \leq k \leq 1.0$. [16]

A large inertia weight (w) eases the local when its value is large and eases the global search when its value is small. To summarized PSO technique in steps it can be consist of just three steps:

1) Calculate fitness for all particles.
2) Update local and global bests.
3) Update velocity and position for all particles.

7. Proposed Method: D* Algorithm Based on PSO

The path planning algorithm which has been used, is the heuristic D* algorithm that depends expansion nodes, in an environment with moving objects. Constructing the environment includes putting the map limitation, start and target locations and the obstacles locations and dimensions.

After constructing the environment, the algorithm will start its work. The robot firstly stands on the target node and start 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 while the other nodes will go to the open list. Taking into consideration the obstacles having a huge cost function and sorting in the closed list from the beginning and the process is completed like that until reaching the start node. When the search is completed the robot will move from the start on the nodes that have been saved in the closed list to the target node (therefore it is called backward searching method) avoiding the obstacles nodes, the current node, and the next step movement node.

The velocity of the moving obstacle is constant and the velocity of the mass point is changing over time (MS × T_s) and it is calculated by:

$$\mathrm{v_x(t)} = \frac{\Delta \mathrm{x\,(t)}}{\mathrm{MS \times T_s}} \quad \ldots (10)$$

$$\mathrm{v_y(t)} = \frac{\Delta \mathrm{y(t)}}{\mathrm{MS \times T_s}} \quad \ldots (11)$$

Where Δx represents the change of distance in X-axis, Δy represents the change of distance in Y-axis, MS represents the motion sample 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 proposed planning for the mass point in dynamic environment is shown in figurer (3). 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 (birdsnumber) 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 bestchosen points from the PSO optimization technique to generate 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 is the next step. The 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 flowchart in figure (4) explains the entire process.

The cubic polynomial will connect the start point and goal point through the waypoints that are used for calculating the following cost function, which will be minimized by PSO algorithm. The final optimal path length represented in the equation (12) below.

Cost $=\sum_{i=1}^{m}\sqrt{(x_i-x_{i+1})^2+(y_i-y_{i+1})^2}$ … (12)

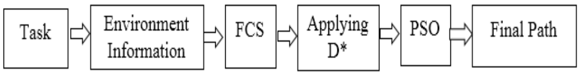


Fig. 3. Proposed planning for the mass point in dynamic environment.

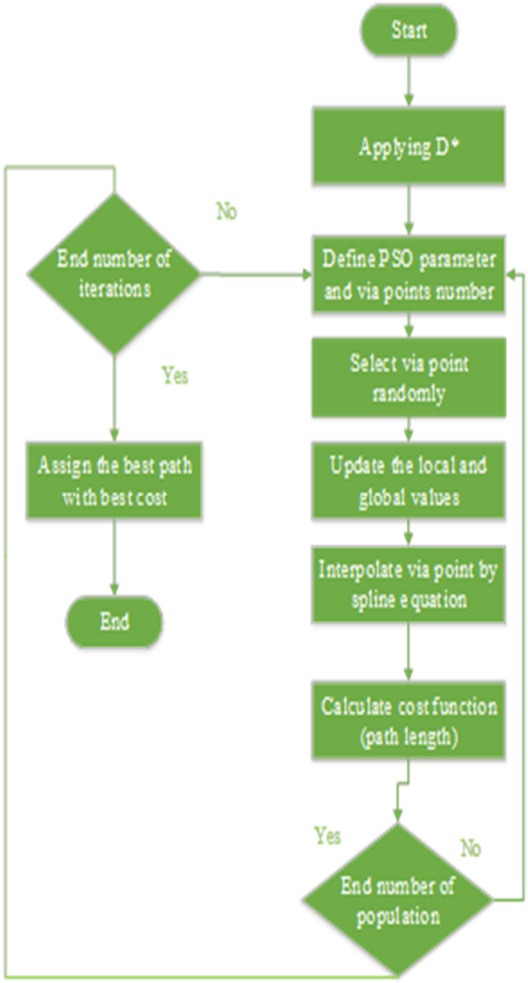

Start
Applying D*
No
End number of iterations
Define PSO parameter and via points number
Select via point randomly
Yes
Assign the best path with best cost
Update the local and global values
Interpolate via point by spline equation
End
Calculate cost function (path length)
Yes
No
End number of population

Fig. 4. The heuristic D* Via Point flowchart for known obstacle position dynamic environment.

8. Simulation Results and Discussion

8.1 First Environment

Here it is applied the two-Link robot arm as a case study. The first and second link length is equal to the half of the robot workspace and the arm has 360 degrees. Figure (5), clarifies the environment analysis and the obstacle motion behavior with four moving obstacle two from these obstacles move in a square motion shape, For a while, this generated path looks like a little bit simple, but this environment featured a critical path, as shown in figure (6), where figure (6.a) represents the elbow down analysis, figure (6.b) represents the elbow up analysis and figure (6.c) represents the elbow up and down analysis. In FCS analysis figures the colored area (blue, red or green) indicates the free and safe 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 completely safe path inside the permitted motion area. Since the FCS is very limited because of the range of obstacle's motion, so the robot will move in elbow down free space as mentioned in figure (2) to the configuration of the two-link arm elbows. Figure (7) shows the two-link robot arm environment which looks very crowded for moving in. Where the process is off-line, the PSO optimization technique is used to get rid of sharp edge and shorten the path and as shown in figure (8), the D* path based on PSO was decreased by 3.2846 cm long from the original path as shown in table (2), which clarify the environment information such as the original D* path length and the new optimal path length which is clearly shorter. Figure (9) represents the new path of D* based on PSO. The joint angles and change in angles based on PSO path are shown in figure (10) and (11) respectively.

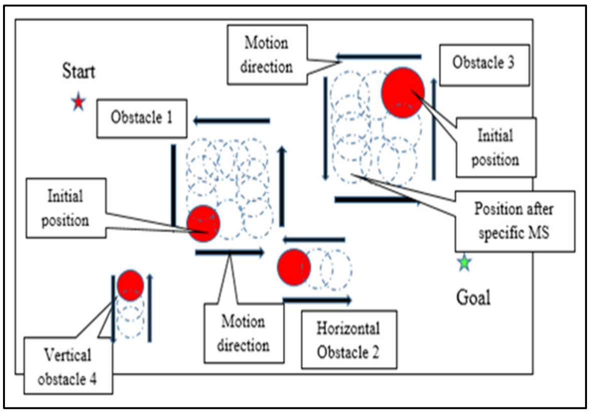
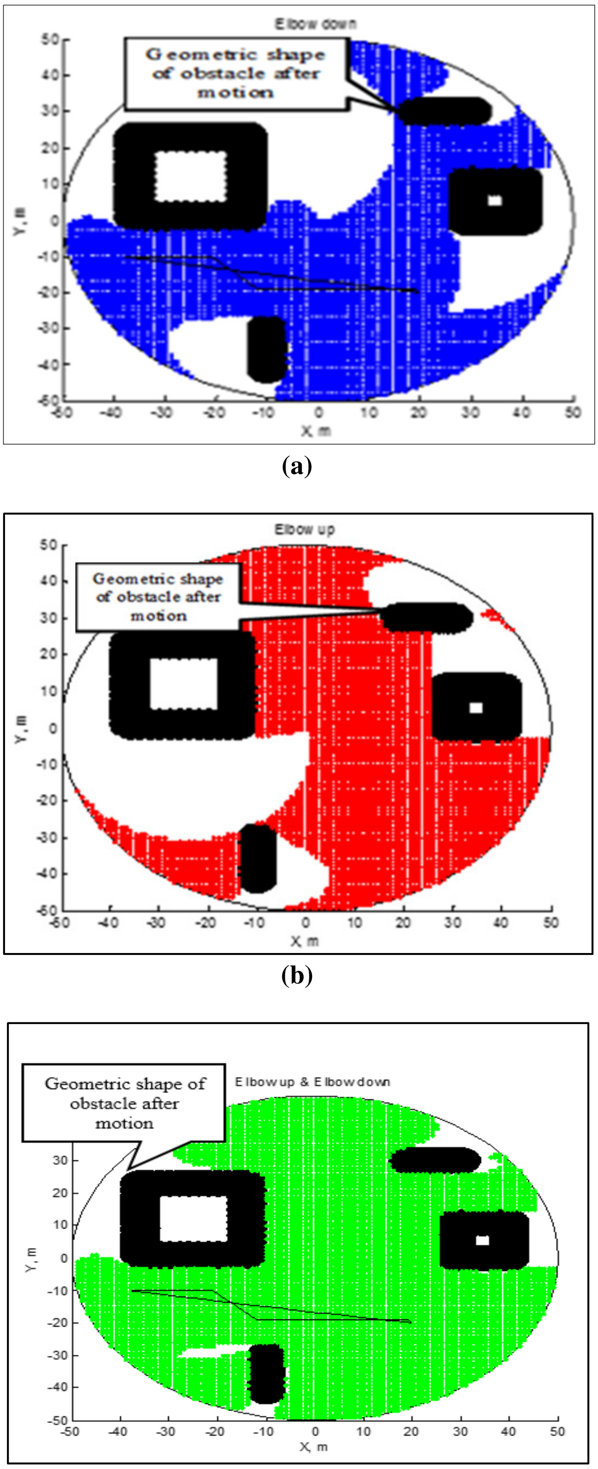


Fig. 5. Obstacle motion behavior.

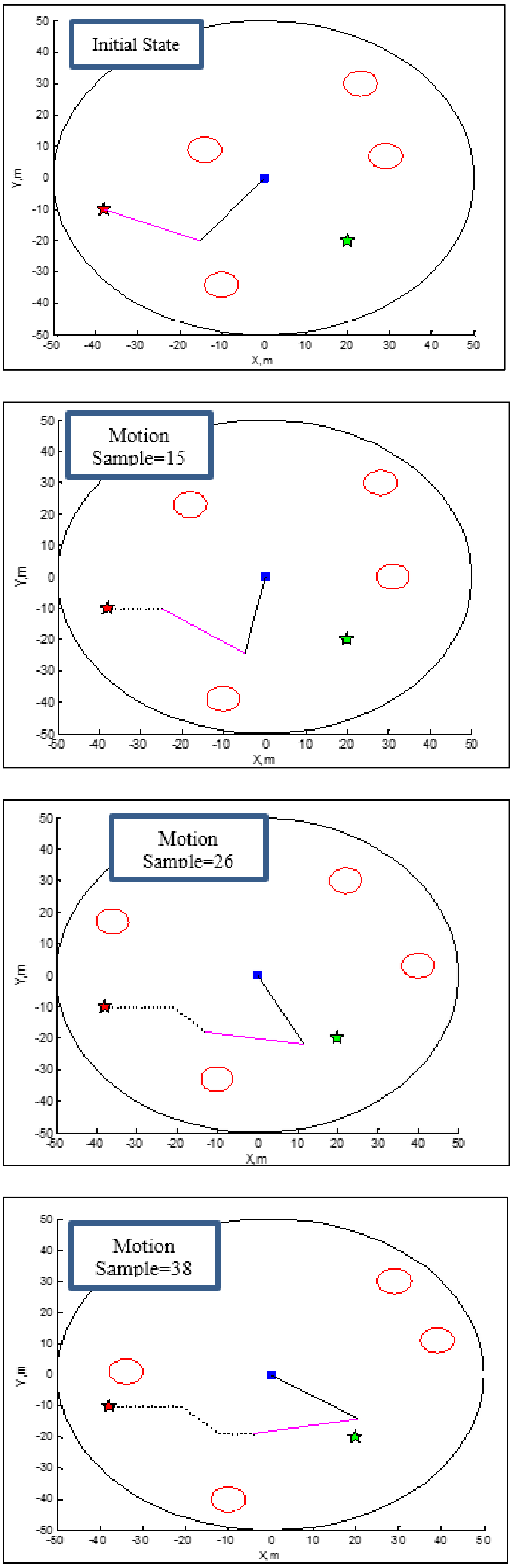


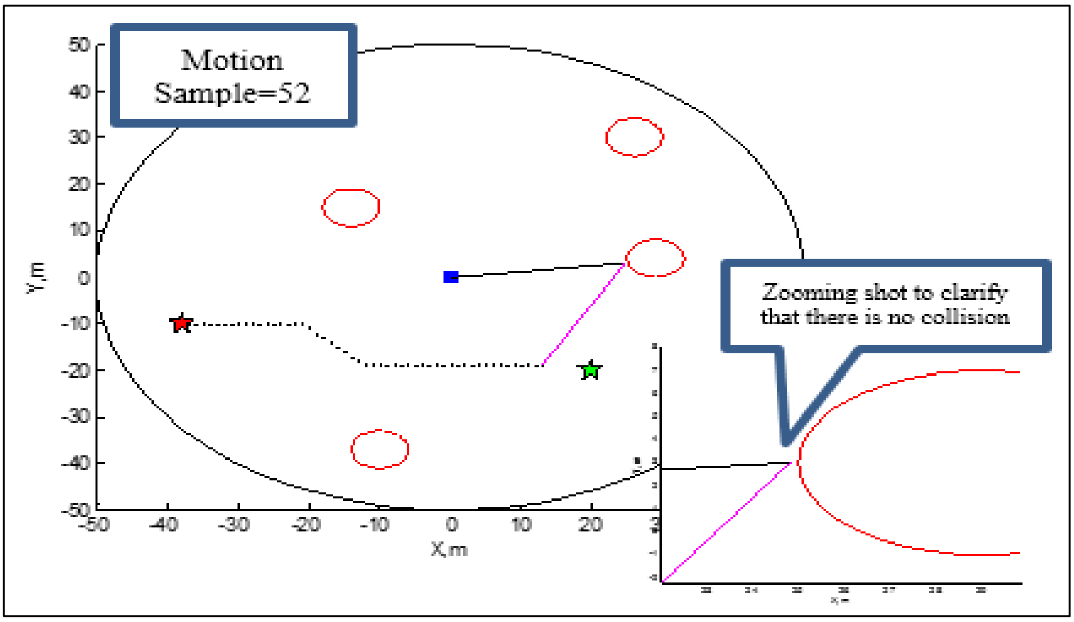
(a)

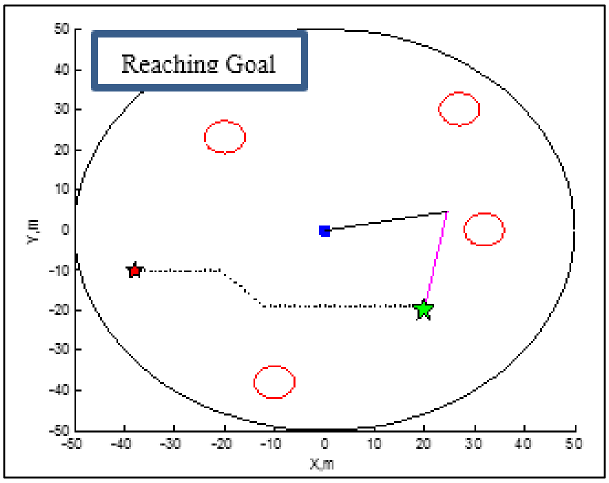
(b)

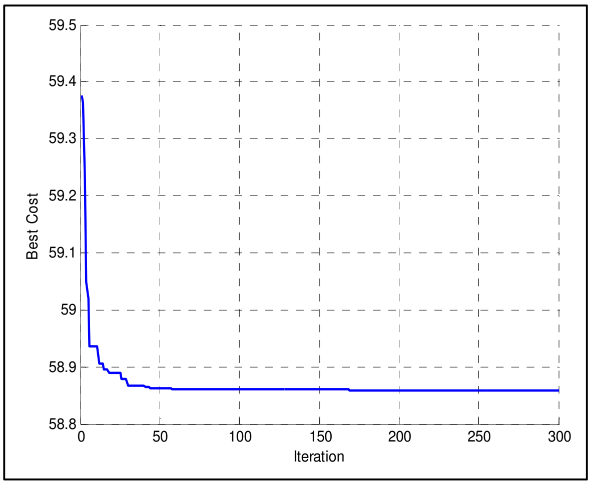
(c)

Fig. 6. FCS for known obstacle position dynamic environment where (a) represents elbow down, (b) represents elbow up and (c) represents the gathering between elbow up and elbow down.





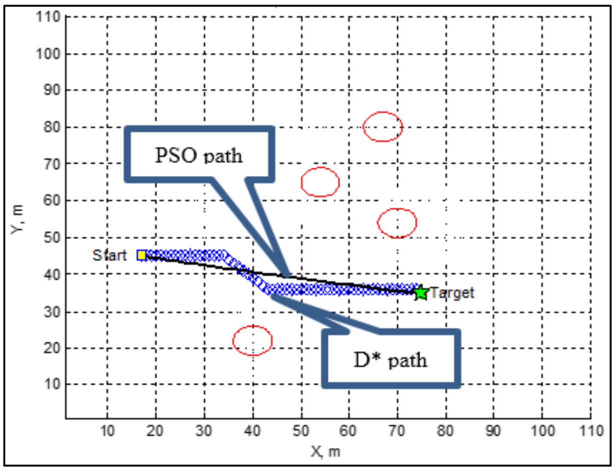




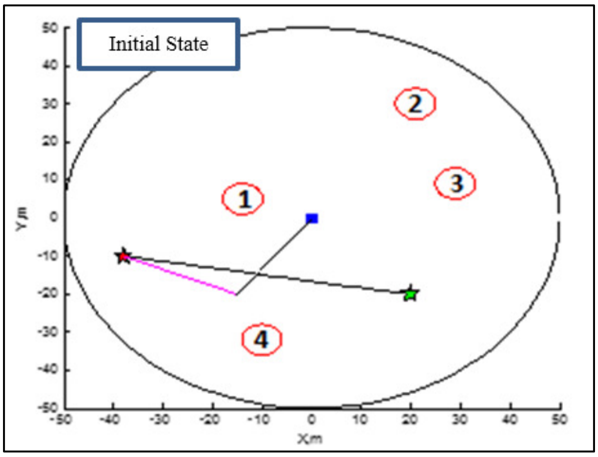
(b)

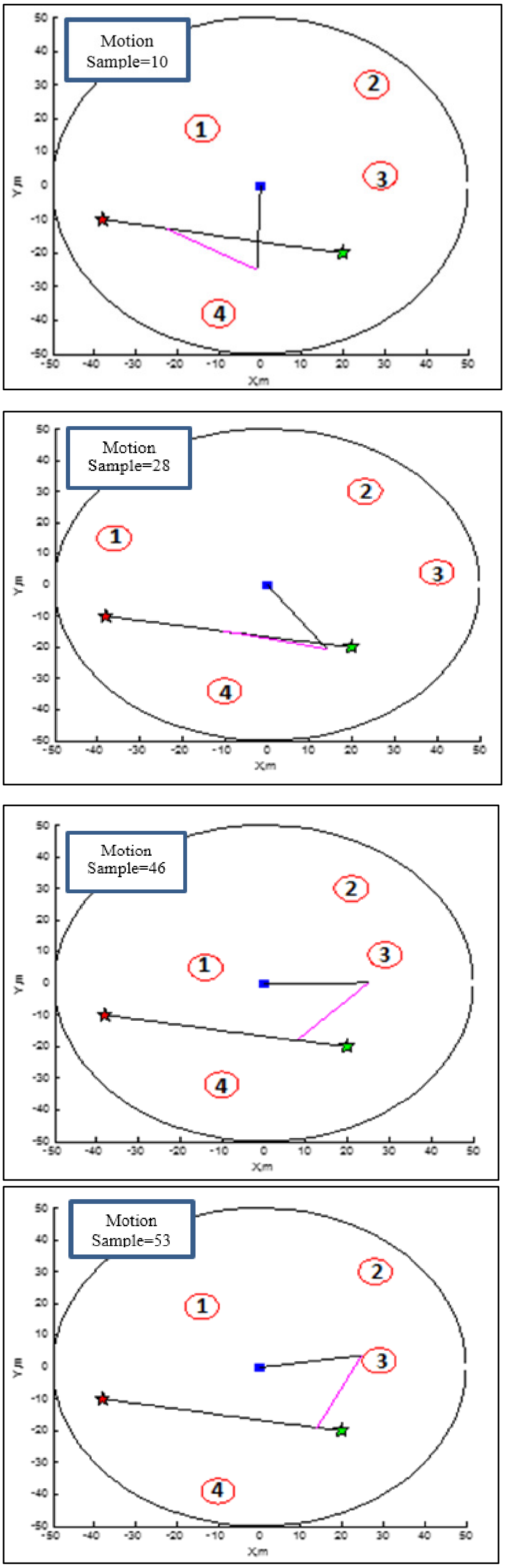
Fig. 7. The two-link robot arm result for the D* path.

Fig. 8. (a) Represents final D* and (b) represents PSO path and the path cost function.



(a)





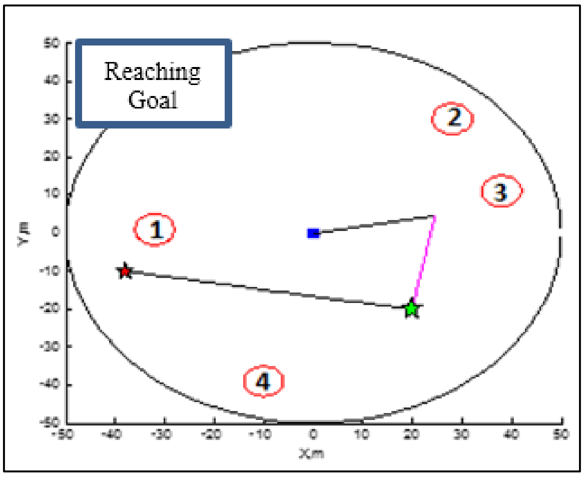


Fig. 9. The two-link robot arm result for the D* based on PSO path.

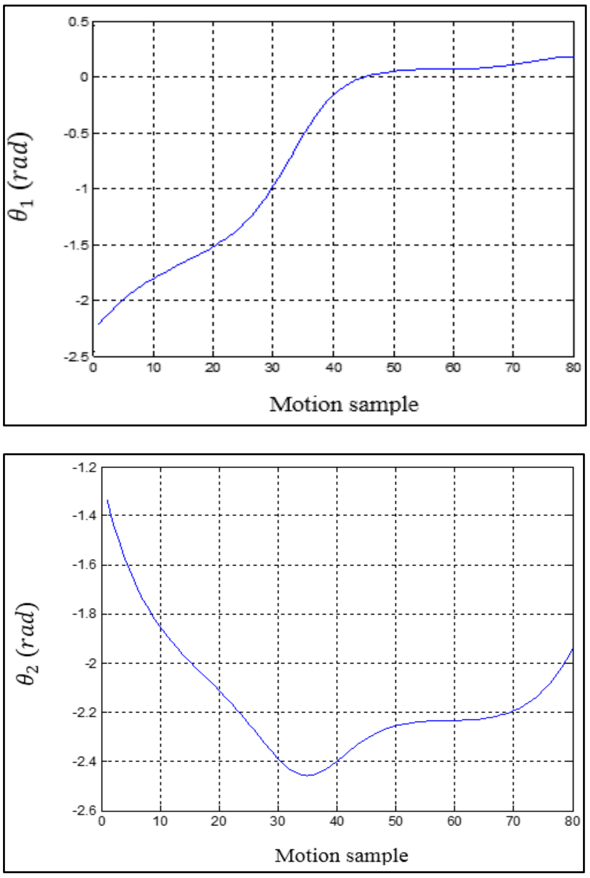


Fig. 10. The first and second angles of the two-link robot arm result for the D* path.

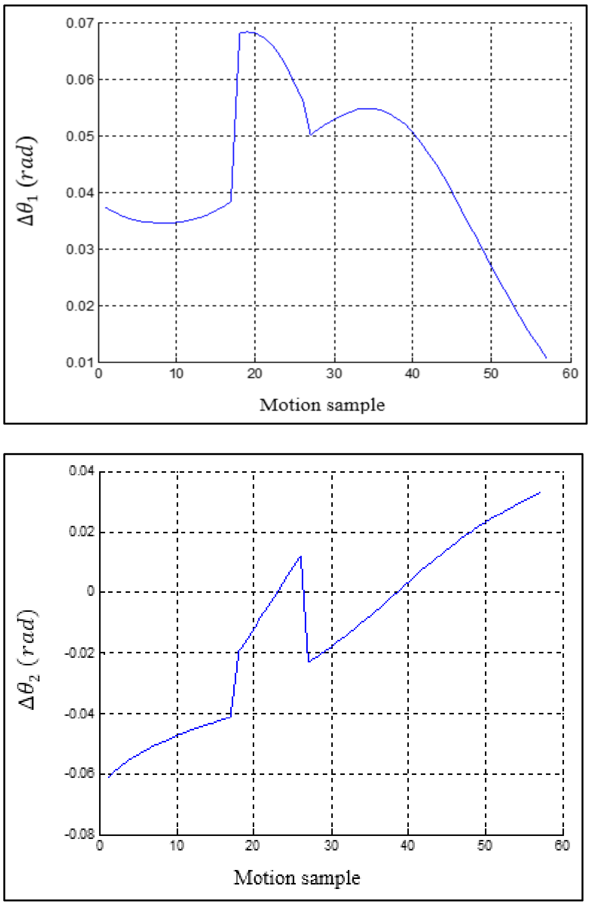
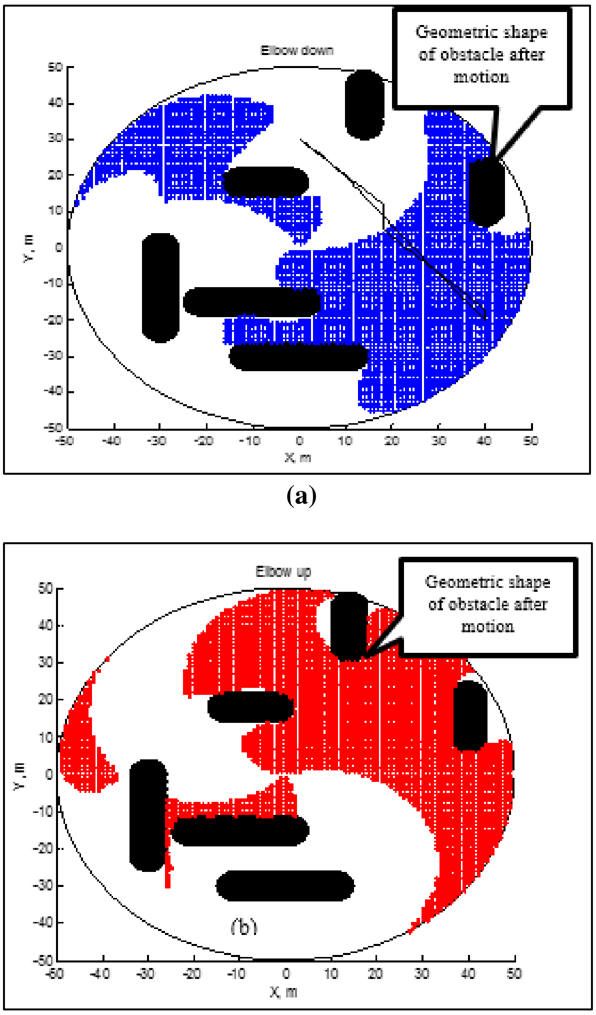


Fig. 11. The change in first and second angles of the two-link robot arm result for the D* path.

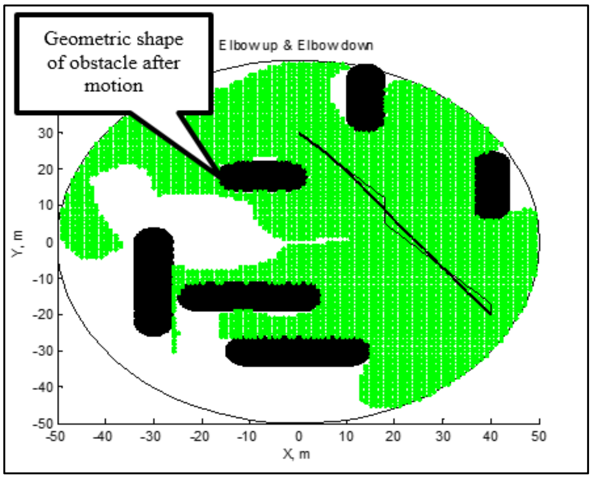
8.2 Second Environment

For more verification, a second more complicated environment with six moving obstacles in such different way in comparing with the first environment and more close to the robot links. Figure (5), also could clarify the environment analysis since this environment has only horizontal and vertical motion styles. This environment featured planning in complex workspace and generate path in narrow area, as shown in figure (12) where figure (12.a) represents the elbow down analysis, figure (12.b) represents the elbow up analysis and figure (12.c) represents the elbow up and down analysis, In FCS analysis figure the colored area (blue, red or green) indicates the free and safe 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 completely safe path inside the permitted motion area. Since the FCS is very limited because of the range of obstacle's motion, so the robot will move in elbow up free space as mentioned in figure (2) to the configuration of the two-link arm elbows. Figure (13) shows the two-link robot arm environment which looks very crowded for moving in. Where the process is off-line, the PSO optimization technique is used to get rid of sharp edge and shorten the path and as shown in figure (14), the D* path based on PSO was decreased by 2.4413 cm long from the original path as shown in table (2), which clarify the environment information such as the original D* path length and the new optimal path length which is clearly shorter. Figure (15) represents the new path of D* based on PSO. The joint angles and change in angles based on PSO path are shown in figure (16) and (17) respectively.



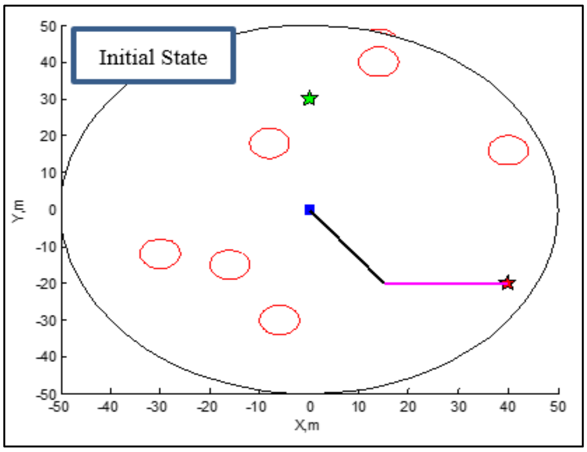
(a)

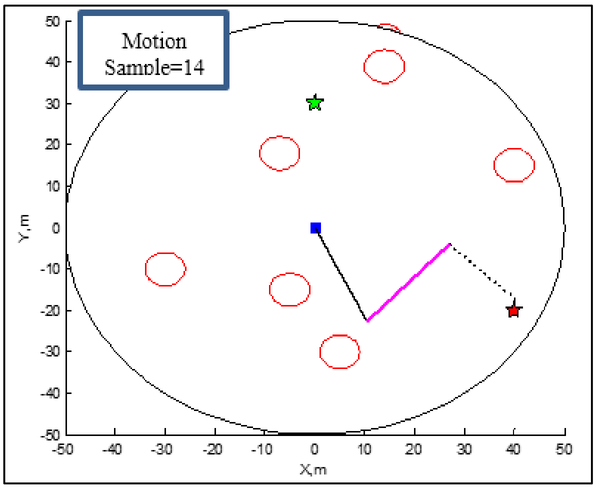
(b)

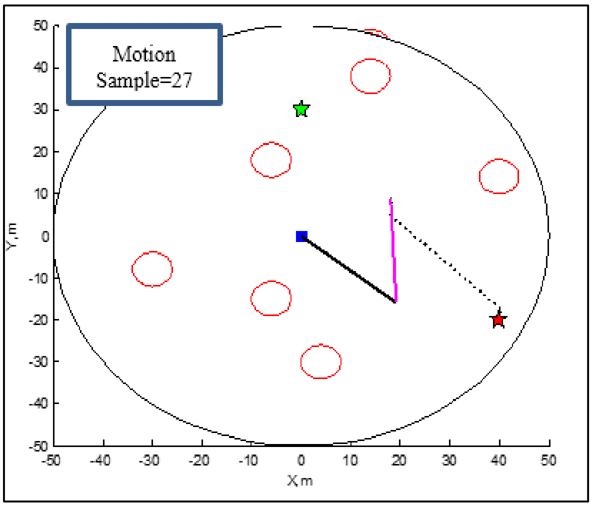


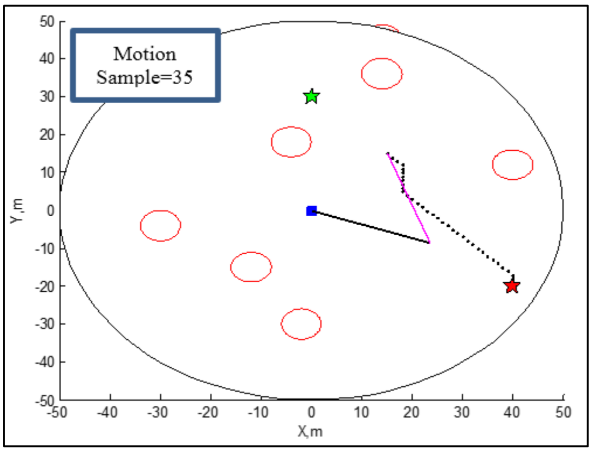
(c)

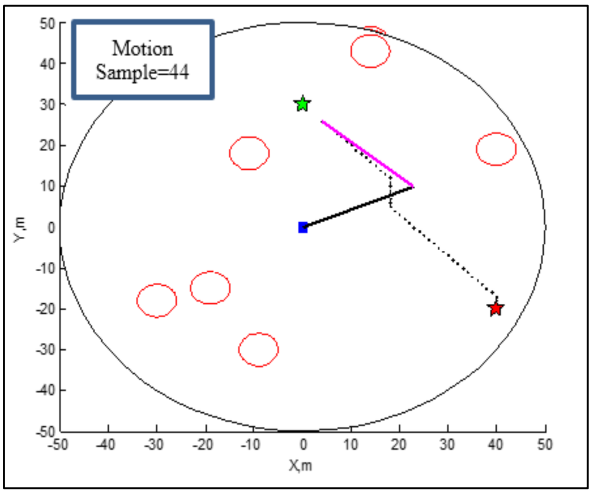
Fig. 12. FCS for known obstacle position dynamic environment where (a) represents elbow down, (b) represents elbow up and (c) represents the gathering between elbow up and elbow down.











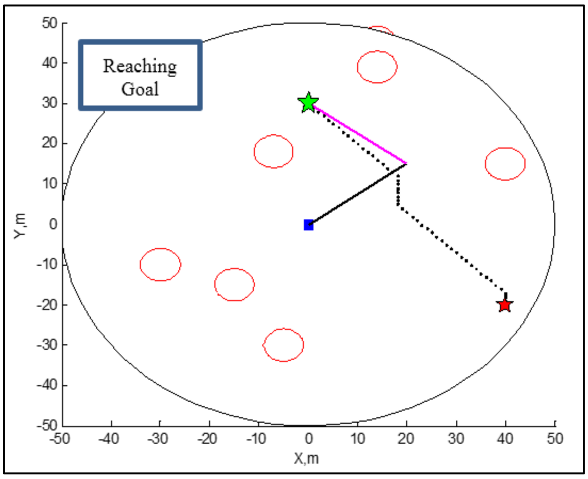
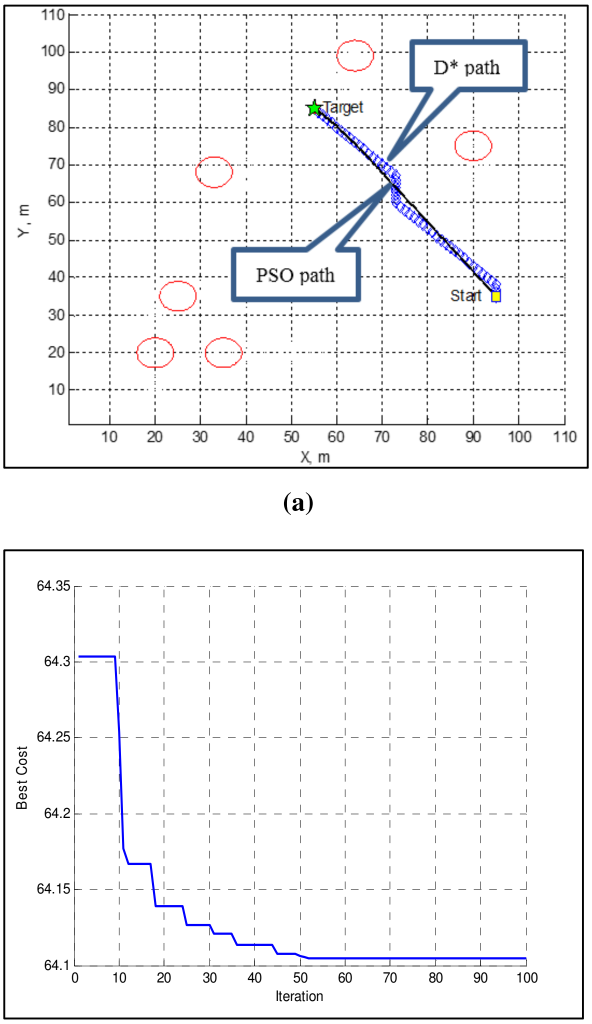


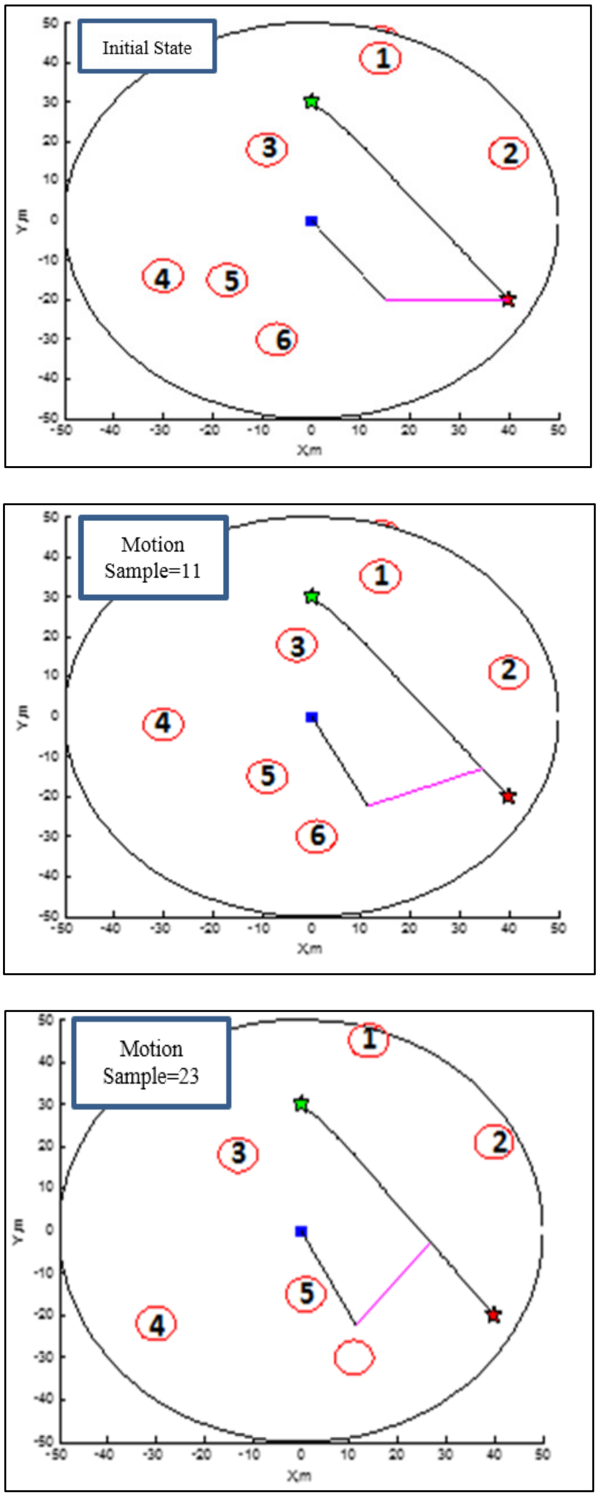
Fig. 13. The two-link robot arm result for the D* path for the second environment



(a)

(b)

Fig. 14. (a) Represents final D* and (b) represents PSO path and the path cost function for the second environment.



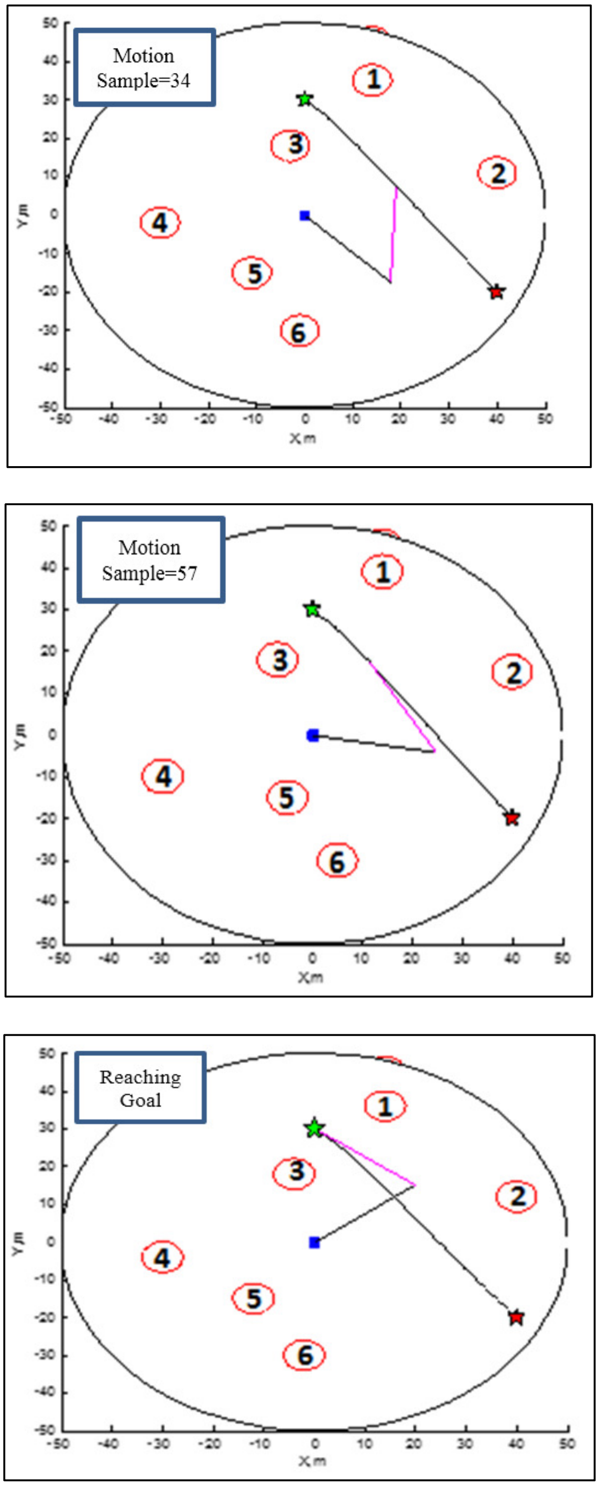


Fig. 15. The two-link robot arm result for the D* based on PSO path for the second environment.

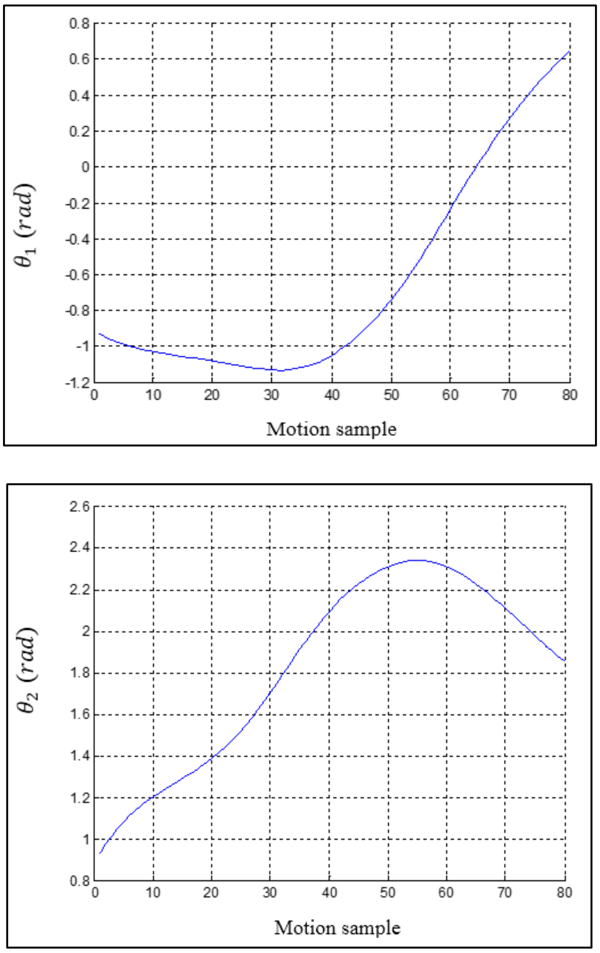
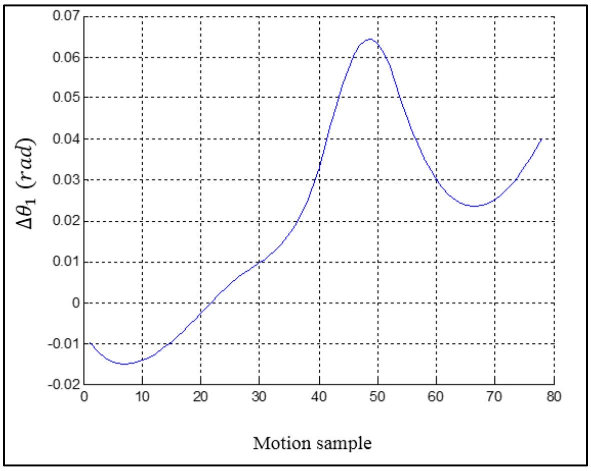


Fig. 16. The first and second angles of the two-link robot arm result for the D* based on PSO path for the second environment.



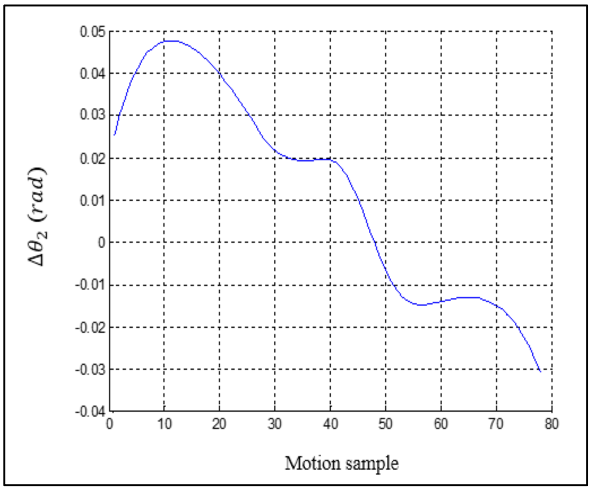


Fig. 17. The change in first and second angles of the two-link robot arm result for the D* based on PSO path for the second environment.

At first, the D* algorithm is applied to find the shortest and best path then the Particle Swarm Optimization method is used to enhance the final path by removing the sharp edges as clearly shown in results figures of the first and the second environment till finding optimal path. The result clearly proves the advantage of using the proposed mixed approach as shown in table (2). In comparison with other path planning algorithms like artificial potential field (APF) [2, 17], the heuristic D* is more suitable for the dynamic environment because of its behavior in saving cost and gives safe and better path length cost function. In addition, the most important issue is the heuristic methods does not suffer from the local minima problem that appears mainly in APF path planning approaches.

Table 2,
Environments information.

Environment	1	2
Obstacles number	4	6
Population	50	50
iteration	300	100
D* path	62.1421 cm	66.5685 cm
PSO path	58.8575 cm	64.1272 cm

9. Conclusions

This paper proposed a methodology that considers solving the path planning problem for two-link robot arm in known position changes dynamic environment utilizing the use of heuristic D* algorithm for finding the best path and the use of PSO optimization technique for getting the final optimal path. From this approach, it was possible to get an acceptable path in spite of the environment difficulties where a completely collision free path could be found at the expense of optimality. Therefore, the PSO optimization technique has been used to reach optimality. The length of the new path produced by the optimization technique was smooth, safe, optimal and shortest which traditional D* method could not find such a path.

10. References

[1] H. Miao, Y.-C. Tian and Y.M. Feng, “Robot Path Planning in Dynamic Environments using a Simulated Annealing Based Approach”, Faculty of Science and Technology Queensland University of Technology, 2014.

[2] F. A. Raheem and M. M. Bader, “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, 2017.

[3] Kikuo Fujimura, “Motion Planning in Dynamic Environments”, Springer-Verlag Tokyo, 1991. https://doi.org/10.1007/978-4-431-68165-6

[4] A. Stentz, The Focused D* Algorithm for Real-Time Replanning. Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, 20-25 August 1995, 1652-1659, 1995.

[5] Enric Cervera I Mateu, “Perception –Based Learning for fine motin planning in robot manipulation”,Jaume-I University of Castello. Spain, 1997.

[6] Z. Shiller, F. Large and S. Sekhavat, “Motion Planning in Dynamic Environments: Obstacles Moving Along Arbitrary Trajectories”, Mechanical and Aerospace Engineering, University of California, Proceedings of the IEEE International Conference on Robotics & Automation Seoul, Korea. May 21-26, 2001.

[7] R. S. M. Asghari and O. A. Tabataba , “Mobile Robot Path Planning by RRT* in Dynamic Environments”, MECS, April 2015.

[8] B. Asadi, “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] A. Vemula, “Safe and Efficient Navigation in Dynamic Environments,” Carnegie Mellon University, Pittsburgh, no. July, 2017

[10] Mark W. Spong, Seth Hutchinson, M. Vidyasagar “Robot Modeling and Control” united states of America first edition, Wiley, Hoboken, 2006.

[11] John J. Craig “Introduction to Robotics Mechanics and Control” United States of America third edition, Pearson Prentice Hall Pearson Education, Inc., Upper Saddle River, NJ. 2005.

[12] A. T. Sadiq, F. A. Raheem, and N. A. F. Abbas, “Optimal Trajectory Planning of 2-DOF Robot Arm Using the Integration of PSO Based on D * Algorithm and Cubic Polynomial Equation,” First Int. Conf. Eng. Res., no. March, pp. 458–467,2017.

[13] A. T. Sadiq, F. A. Raheem, and N. A. F. Abbas, “Robot Arm Path Planning Using Modified Particle Swarm Optimization Based on D * Algorithm” Al-Khwarizmi Engineering Journal, Vol. 13, No. 3, pp. 27-37, 2017.

[14] P. Boonyaritdachochai, et al., "Optimal congestion management in an electricity market using particle swarm optimization with time-varying acceleration coefficients," Computers & Mathematics with Applications, vol. In Press, Corrected Proof, 2010.

[15] Poli, R., Kennedy, J. and Blackwell, T. Particle Swarm Optimization: An Overview. Swarm Intelligence, 1, 33-57, 2007

[16] K. T. Chaturvedi, et al., "Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch," International Journal of Electrical Power & Energy Systems, vol. 31, pp. 249-257, 2009.

[17] Ni, J.J., Wu, W.B., Shen, J.R. and Fan, X.N. 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

تخطيط مسار ذراع روبوت باستخدام خوارزمية D^* بالاعتماد على تقنية سرب الطيور

أُمنية ابراهيم حميد** فراس عبد الرزاق رحيم*

*،** قسم هندسة السيطرة والنظم / الجامعة التكنولوجية / بغداد / العراق

البريدالالكتروني: 60124@uotechnology.edu.iq

البريدالالكتروني: 61126@uotechnology.edu.iq

الخلاصة

ان ايجاد حل للمسار في بيئة ديناميكية يمثل تحديا للباحثين في مجال الروبوتات فضلاً عن ذلك فأنها تمثل القضية الرئيسية للروبوتات واذرع الروبوتات التي يتطلع اليها العالم اليوم. يعد المسار الخالي من التصادم في البيئات ذات العوائق المتحركة مثل الكائنات المختلفة والبشر والحيوانات أو روبوتات أخرى مشكلة حقيقية ويحتاج إلى حل. فضلاً عن ذلك، الحدود الدنيا المحلية والحواف الحادة هي المشكلات الأكثر شيوعًا في جميع خوارزميات تخطيط المسار. الهدف الرئيس من هذا العمل هو التغلب على هذه المشكلات باستخدام خوارزمية تخطيط المسار *D بالاعتماد على تقنية التحسين PSO ، علاوة على ذلك، يركز هذا العمل على الجزء الحسابي من تخطيط الحركة في بيئة ديناميكية متغيرة تمامًا مع كل motion sample . نظرًا لأن نوع البيئة الذي تمت مناقشته هنا هو البيئة الديناميكية المعروفة فأن الحل سيكون off-line. ان الميزة الرئيسة لتخطيط المسار الـ off-line هو دائما الحصول على حل المسار ال global الامثل. ، والتي هي قادرة على التغلب على جميع الصعوبات الناجمة عن السلوك الديناميكي للعقبات. ولذلك تم اللجوء الى خواروميه *D مع تقنية التحسين الPSO حيث تم استخدام خوارزمية تخطيط المسار *D للحصول على اقصر مسار ومن ثم تطبيق تقنية تحسين المسار PSO لتحسين المسار النهائي. تم استخدام ذراع الروبوت ثنائي درجات الحرية بوصفه حالة دراسية لاختبار الطريقة المقترحة وهي تمثل حالة اكثر صعوبة من ال mobile robot. كما ان نتائج المحاكاة تظهر فعالية الطريقة المقترحة والتي تظهر بوضوح المسار الامن والقصير تماما.