Al-Khwarizmi Engineering Journal,Vol. 13, No. 3, P.P. 27- 37 (2017)

Al-Khwarizmi Engineering Journal

Robot Arm Path Planning Using Modified Particle Swarm Optimization based on D* algorithm

Ahmed T. Sadiq* **Firas A. Raheem**** **Noor Alhuda F. Abbas*****

*,****Department of Computer Science/ University of Technology*

**Department of Control and System Engineering/ University of Technology*

Email: Drahmed_tark@yahoo.com

Email: dr.firas7010@yahoo.com

Email: Nooralhudafabbas92@yahoo.com

(Received 6 September 2016; accepted 7 February 2017)

https://doi.org/10.22153/kej.2017.02.001

Abstract

Much attention has been paid for the use of robot arm in various applications. Therefore, the optimal path finding has a significant role to upgrade and guide the arm movement. The essential function of path planning is to create a path that satisfies the aims of motion including, averting obstacles collision, reducing time interval, decreasing the path traveling cost and satisfying the kinematics constraints. In this paper, the free Cartesian space map of 2-DOF arm is constructed to attain the joints variable at each point without collision. The D*algorithm and Euclidean distance are applied to obtain the exact and estimated distances to the goal respectively. The modified Particle Swarm Optimization algorithm is proposed to find an optimal path based on the local search, D* and Euclidean distances. The quintic polynomial equation is utilized to provide a smooth trajectory path. According to the observe results, the modified PSO algorithm is efficiently performs to find an optimal path even in difficult environments.

Keywords: *D*, Free Cartesian Space, Path Planning, Particle Swarm Optimization (PSO), Robot Arm.*

1. Introduction

A robot arm is spontaneously controlled manipulator which possesses programming capacity into three or more axes with multipurpose uses in various automation implementations. The main issue for attaining autonomous robots arm is the planning for a collision-free path. However, in environments that comprise static or moving obstacles including human workers and other robots, path planning concerns in finding the possible path from the initial to the target configurations, providing that no point in the motion trajectory nor on the links of the arm collides with any obstacle, and ensures the shortest path [1,2]. With ideal system of path planning, navigation of robot arm can be achieved on its own with no human interference [3].

The path can be constructed either in the joint space or Cartesian space. In the joint space, the path is particularly identified for each distinct joint, however, the end-effector Cartesian position is only predicted at the start and target positions. Moreover, the Cartesian path has the characteristic of being easily specified, and the manipulator motion is entirely identified [4].

Depending on the availability of information related to the robot arm environment, the path planning is subdivided into global and local path planning. The global path planning has the whole required information, in contrast, the local path planning has partially or entirely anonymous information [3].

Motion planning possesses many algorithms that can be categorized into classical and heuristic. Artificial potential field, visibility graph, and cell decomposition and many others are various forms of the classical approaches. Each one faced the problems of local minimum stacking and has

elevated account of cost and time. On the other hand, the heuristic techniques are mainly utilized for their ability to solve the problem of path planning, and offer many advantages such as easiness in implementation and swift generation of various approved solutions [3, 5].

Particle swarm optimization is a heuristic technique. Simplicity, rapid convergence, power and impose few tuned parameters are the main features that differentiated it on the other complex optimization algorithms [5].

In this paper, the modified PSO algorithm is proposed depending on cost of D* and Euclidean distance to find an optimal path of two links arm that moved in 2D.

2. A Review of Previous Researches

In order to solve robot navigation problem, numerous methods have been proposed by many researchers. Recently, a series of intelligent ideas are used, such as genetic algorithm, ant colony optimization and particle swarm optimization due to their ability for simultaneous calculations and efficient solving the navigation problems. Guo et al [6] proposed the Cultural based PSO (CBPSO) algorithm which is to find the collision-free trajectory for redundant robot manipulators in the presence of fixed obstacles. The authors used Quadrinomial and quintic polynomials to describe the segment of the trajectory. Rokbani and Alimi [7] applied the Inverse Kinematics PSO (IK-PSO) to find the inverse kinematics of robot manipulator. The objective is to generate the trajectory path by forward Kinematics depending on the joint variables that obtained from IK-PSO. Kim and Lee [8] proposed the PSO algorithm based on a normalized step cost (NSC) for trajectory optimization. A NSC approach was considered as new method for initializing the particles in PSO .The optimal trajectory is the one that had a lower cost and converged faster, without any collision with the obstacle in the workspace.

3. Kinematics Modeling of Two-Link Robot Arm

Kinematics is Geometry of Motion. kinematic problems can be classified into two types, one of them is the forward kinematic problem which is to allocate the position of end effector and its orientation through the provision of the values for the robots' joint variables, while the second problem is inverse kinematic which is specified all possible sets of joint angles and link geometries through provision the value of end effector position and orientation. The general purpose of using inverse kinematics is to obtain joint space through a given Cartesian space (position and orientation of a manipulator end-effector) [9, 10].

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 [11, 12].

Consider the diagram of Fig. (1). the following equations are used to compute $\theta_1$and θ_2 [11, 12, 13].

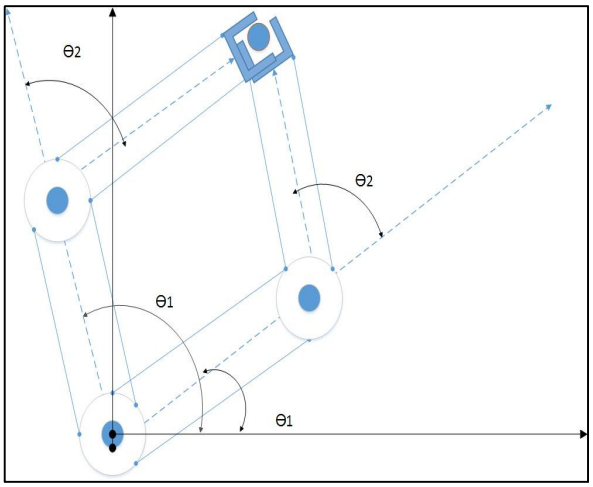


Fig. 1. Two-link arm.

$$\cos\theta_2 = \frac{p_x^2+p_y^2-l_1^2-l_2^2}{2l_1l_2} = D \quad \dots (1)$$

Since, the sin θ_2 is

$$\sin\theta_2 = \mp\sqrt{1-D^2} \quad \dots (2)$$

The two possible solutions for θ_2 can be obtained by writing as:

$$\theta_2 = \tan^{-1}\left(\frac{\mp\sqrt{1-D^2}}{D}\right) \quad \dots (3)$$

Finally, $\theta_1$can be found by the following equation:

$$\theta_1 = \tan^{-1}\left(y/x\right) - \tan^{-1}\frac{l_2\sin\theta_2}{l_1+l_2\cos\theta_2} \quad \dots (4)$$

Where l_1 and l_2 are length of first and second links respectively [11, 12, 13].

The main advantage of this approach is to recapture both of the elbow-up and elbow-down solutions by choosing the negative and positive signs in the Equation [13, 14, and 15].

The inverse kinematics can be considered as the first step for generate the free Cartesian space on it the path planning algorithm is searching to find optimal path.

4. Free Cartesian Space Analysis

In this paper, workspace space has been compiled and analyzed based on the inverse kinematics result. The free Cartesian space of Two-link arm can be defined as space with set of all points that can be reached by specific end-effector configurations. These points associated with joint angle (θ_1 andθ_2) which has been obtained from inverse kinematics. In environment contains many obstacles, the shape and volume of free Cartesian space are varied according to the shape, size, position, and number of obstacles in addition to mechanical limits of joints (for the proposed method, modeling and simulation as in table 1). These constraints influence and restrict motion of manipulator as well as separate the workspace into reachable area and unreachable area.

Table 1,
Range of two joints for arm's links.

Link Number	Range of Arm's Joint In Degree
Joint 1	$0 \leq \theta_1 \leq 360$
joint 2	$-90 \leq \theta_2 \leq 90$

Computation of the free Cartesian space has been done by analyzing points in environment and finding all possible solution of point's configuration after checking both limitation of joint and collision with obstacles. The checking function is varied according to shape of the obstacle (polygon, circle). In polygon obstacle it is dependent on vertices positions of obstacle but in circle obstacle it is depending on center point as well as radius. Moreover, there are three possible cases for each coordinates point in Cartesian space. In the first case, the point has two solutions (elbow up and elbow down configurations) as shown in Fig. (2). But in the second case, the point has just one solution either the manipulator is fully extended to reach a point at the border of manipulator's reachable workspace, or point has one configurations (elbow up or elbow down) due to the other configuration collide the obstacle at any part of manipulator links. while the in the third case, the point has no solution when coordinates point are out of reach of manipulator or in obstacles area, or the two configurations collide with obstacle at any part of arm links.

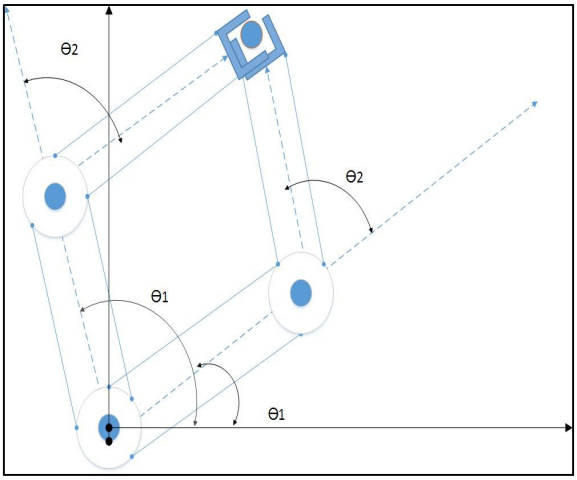


Fig. 2. Two-link arm elbow up and elbow down configurations.

The results of analysis points have been formed as three spaces, the first space is representing all points that reachable in elbow up solution (free elbow up space), and the second space is representing all points that have elbow down solution (free elbow down space), and the third space is unreachable space which including points outside the manipulator reachable workspace and points that collide with obstacle and points in obstacle area. Moreover, the free Cartesian space is comprising of all points have at least one solution (elbow up solution space and elbow down solution). The length of link of arm is also can be considered as major factor to construct the free Cartesian space.

5. Particle Swarm Optimization

PSO is technique of stochastic optimization, which operates on a principle that simulates the social behavior of the bird flocking. This technique are used in many applications and fields to recognize the parameters required for minimizing or maximizing [16, 17]. In a PSO system, the individuals' swarm that are flying through the search space called particles and each particle is considered as a candidate optimal solution for the problem. The best position visited by a particle is influencing the particle position and the position of the best particle in its neighborhood i.e. its own and neighboring particles experiences respectively [16, 17].

Various PSO models have been evolved depending on the topology of neighborhood, namely: local best (lbest) and global best (gbest). When the whole swarm represents the

neighborhood of a particle, the global best position of the particle is considered as the best position in the neighborhood, in this case the gbest PSO is the resulting algorithm. While the lbest PSO algorithm can commonly be assigned when smaller neighborhoods are used, and the best particle is tracked by each particle which is gained by the particle local topological neighborhood. The fitness function is used to measure the performance or the activity of each particle. This function is differed according to the optimization problem. Each particle in the swarm has characteristics as follows: (pos_{ij}) is referred to the particle current position, (vel_{ij}) is referred to the particle current velocity and ($pbest_{ij}$) is referred to the particle personal best position. The best position reached by the particle i is referred as the personal best position of particle i, while the best position reached by all particles is referred as the global best position of particles ($gbest_{ij}$) [17, 18, 19].

The PSO algorithm run on the following principle: at first the particles are initialized with randomly distributed position within the workspace. Furthermore the velocity of these particles is also assigned randomly. Each particle has a memory that stores all the best previously visited positions in addition to the fitness function in that position which is improved over time. At each iteration of PSO algorithm, pos_{ij} and vel_{ij} vector of particle is modified over each dimension j by using the equations (5 and 6) in order to guide these particles toward either the personal best vector or the swarm's best vector [2, 17].

$$vel_{ij}^{t+1} = w \times vel_{ij}^{t} + c_1 \times R_1 \times \left(pbest_{ij} - pos_{ij}\right) + c_2 \times R_2 \times \left(gbest_{ij} - pos_{ij}\right) \quad \dots (5)$$

$$pos_{ij}^{t+1} = pos_{ij}^{t} + vel_{ij}^{t+1} \quad \dots (6)$$

Where c_1 and c_2 are the cognitive coefficients (c_1+c_2<=4), $pbest_{ij}$ is the personal best vector, and $gbest_{ij}$ is the swarm's best vector. R_1 and R_2 are random real numbers between 0 and 1, while the inertia weight w controls the particle momentum, it is decreases from (w_{max} = 0.9 to w_{min} = 0.4) over the whole run according to next equation [2, 17]:

$$w = w_{max} - \frac{w_{max} - w_{min}}{\text{iteration}_{max}} * iteration\_num \dots (7)$$

If the value of vel_{ij} is less than vel_{min}or greater than vel_{max} then the corresponding value is replaced by vel_{min} or vel_{max} , respectively. Where vel_{max} is maximum velocity parameter [2, 20, 17, 18].

6. D* Algorithm

The D* also called " Dynamic A*" algorithm is used to analyzed the environment. The arcs cost is dynamically changed at algorithm runs. Each node to be assessed by D*algorithm is preserved in an open list with one of varied conditions (new, open, closed, raise, lower).

D* algorithm is working by backward searching that begins from the goal point until reaching the start point via repeatedly choosing a node from open list for evaluation and computing its cost to the goal. Then this node is expanded to the eight neighboring nodes in order to transmit its changes to these nodes and they eventually placed in the open list with a minimum cost. Nevertheless, in the robot arm environment, only nodes belong to free Cartesian space are evaluated to place them in open list. While, the rest is ignored.

Moreover, every expanded node possesses a back-pointer and precise cost to the goal according to an equation (8).

$$F = g + h \quad \dots (8)$$

Where the F is objective function and g is the estimated cost from star point while h is the cost to the goal. This algorithm uses the heuristic function for increasing and reducing the propagate cost and focus cost respectively [21].

7. Proposed Methodology

To solve the path planning problems for Two-Link arm, the modified lbest PSO algorithm is proposed to generating the collision free path. The lbest PSO algorithm is locally searching through work space (free Cartesian space) depending on D* and Euclidean distance costs to find arm optimal path from its start to goal configurations according to several steps.

In the premier step the start node with initial velocity equal to zero is considered as current position and its eight neighborhood nodes (particles) are determined to be tested by lbest PSO algorithm. In which only the nodes belonging to free Cartesian space are tested by applying the velocity equation (5),

Where $gbest_{ij}$ indicate to the minimum D* cost (at eight tested neighborhood nodes), pos_{ij} indicate to the D* cost at tested node, and $pbest_{ij}$ indicate to the Euclidean distance (E) from tested node to the goal node that is depending solely on the direct orientation to the goal and can be computed for every point in the free Cartesian space according to following equation (9).

$$E = \sqrt{(x_p - x_g)^2 + (y_p - y_g)^2} \quad \dots (9)$$

Where the (x_p, y_p) represent the x-y coordinate of the free Cartesian point and (x_g, y_g) represent the x-y coordinate to the goal point.

The coefficients R_1 and R_2 have significant role to guide the solution either toward the node with minimum D* cost or toward the node with minimum Euclidean distance. Then only one particle with minimum cost and high probability is selected to be at the current node in the proposed path.

in case of the current node does not possess tested neighborhood nodes belong to the free Cartesian space, then the node is considered as dead point and PSO algorithm has to be returned to the last node in the path for selecting another node. While in the next step the new velocity for each particle (potential solution) is set depending on its prior velocity vel_{ij}^{t}. The lbest PSO repeatedly generate and test eight neighborhood nodes for current node until reaching the goal point and the proposed path is considered as the candidate optimal path. This process is iterated until a specific number of iterations are achieved or minimum cost is obtained. Finally the optimal path is formed from entire iterations via selecting the best candidate solution with the minimum cost function which can be calculated by the following equation:

$$\text{Cost}=\sum_{i=1}^{m}\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad \dots (10)$$

Where, m is the number of particles which is equal to the number of nodes along the path, (x_i, y_i) represents the current position of i^{th} particle. Depending on the number of the grid, the optimal path may be more than one path. Each path has the same cost but with differences in some points. All process is done as in the flowchart shown in figure (3).

8. Path Smoothing Using Quintic Polynomial Equation

Trajectory can be defined as path which be followed by a moving manipulator through work space at specific time. Amount of time on which trajectory is carried out can be determined as (t_f- t_0). Since the velocities and accelerations are also parameterized by time, they can be calculated along the trajectories by differentiation with respect to time. The set of Quintic Polynomial Trajectory equations includes a fifth order polynomial position equation, the first derivative is a fourth order velocity equation and the second derivative is a third order polynomial acceleration equation. The trajectory has six constraints which is as initial and final for each one of configurations, velocities and accelerations [14].

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad \dots (11)$$

According to equation (11) at time t_0, the initial position, velocity and acceleration equations are:

$$q(t_0) = q_s \quad \dots (12)$$

$$\dot{q}(t_0) = v_s \quad \dots (13)$$

$$\ddot{q}(t_0) = a_s \quad \dots (14)$$

Where (q_s, v_s, a_s) representing the initial position, velocity and acceleration respectively [14].

While at time t_f the final position, velocity and acceleration equations are:

$$q(t_f) = q_g \quad \dots (15)$$

$$\dot{q}(t_f) = v_g \quad \dots (16)$$

$$\ddot{q}(t_f) = a_g \quad \dots (17)$$

Where (q_g, v_g, a_g) represent the target position, velocity and acceleration respectively [14]. Depending on appropriate number of derivatives for equation (11), the following equations are obtained:

$$q_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 + a_5 t_0^5 \quad \dots (18)$$

$$v_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2 + 4a_4 t_0^3 + 5a_5 t_0^4 \quad \dots (19)$$

$$\alpha_0 = 2\,a_2 + 6a_3 t_0 + 12a_4 t_0^2 + 20a_5 t_0^3 \quad \dots (20)$$

$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \quad \dots (21)$$

$$v_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \quad \dots (22)$$

$$\alpha_f = 2\,a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \quad \dots (23)$$

According to Quintic Polynomial Trajectory Planning, the optimal path that is generated by lbest PSO is smoothed by randomly selected of via points from this path as well as start and goal points. These via points are connecting depending on equations (14, 18, 19, 20 and 21) at (t_f- t_0). The result from this stage is the optimal smoothing path from q_s to q_g.

9. Results and Discussion

In this paper, the proposed method was tested by using Matlab program (2015a) with a difficult proposed map and has a limited from (-50 to 50) cm for both x-y dimensions. The length of

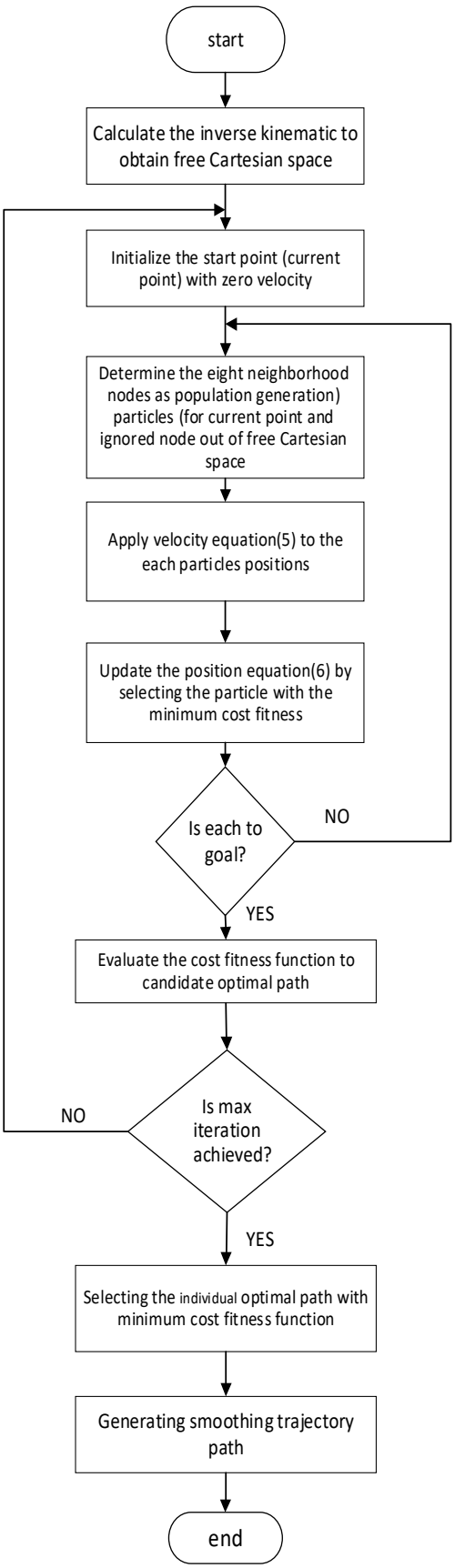
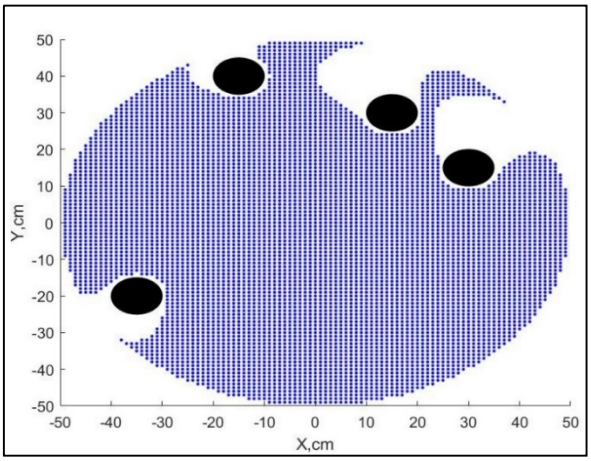
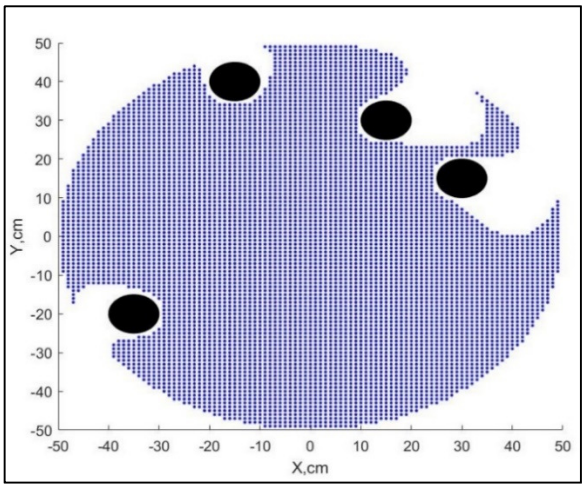


Fig. 3. Flowchart of modified PSO.

The robot links for both link1 and link 2 is proposed equal to 25 cm. Initially, the proposed environment was analyzed and its elbow up and elbow down solutions are computed by using inverse kinematics equations (1, 2, 3, and 4). The free elbow down space is constructed by using the solution of inverse kinematics with the negative sign of equations (3, 4) and ensuring no collision with obstacles, as shown in figure (4, a). While the free elbow up space is constructed by using the solution of inverse kinematics with the positive sign of equations (3, 4) and ensuring no collision with obstacles, as shown in figure (4, b). Ultimately the free Cartesian space map is constructed by all the reachable points in elbow up and elbow down configurations figure (5).



(a)



(b)

Fig. 4. (a): Two-link robot manipulator elbow down configuration solution, (b): Two-link robot manipulator elbow up configuration solution.

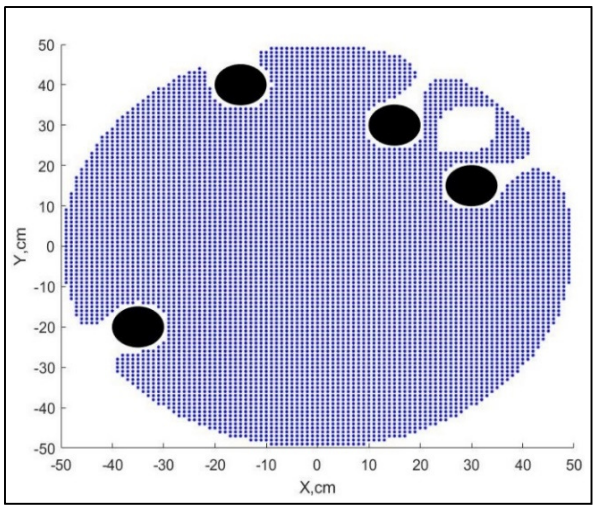


Fig. 5. Two-link robot manipulator free Cartesian space.

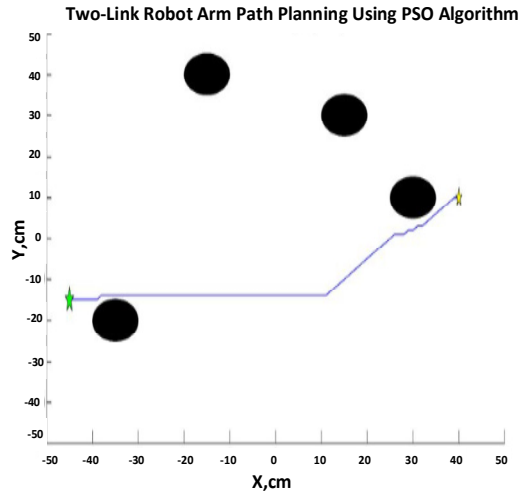


Fig. 6. PSO optimal path solution.

In figures (4, 5), the small dots denoted the free Cartesian space reachable points while the black big circles referred to the proposed obstacles with 5 cm radius, and the white space denoted either unreachable points due to the collision with the obstacle or due to robot joints limit.

It is a crucial that the start and goal task are part of free Cartesian space, otherwise no path can be planned by the arm, therefore, it is of importance to check the initial and target configurations for the arm offline before running the algorithm.

Once the cost functions for each point in free Cartesian space were computed by D* algorithm and Euclidean distance, the modified algorithm was run with the following proposed parameters: initial velocity = 0, the coefficients $c_1 = 3$, $c_2 = 1$ in order to guide Lbest PSO toward the high probability of D* value, initial $w = 0.4$ and max iteration number = 300. In this proposed method, one or more optimal path may be presented. Moreover, these paths have the same cost with difference in some middle points.

The optimal path from the start point (40, 10) to the goal point (-45,-15) with a cost fitness of 92.6232 and executed for 78s is shown in the figure (6).

Figure (7) illustrates the smooth trajectory path that is computed by using the Quintic Polynomial Trajectory equations from $t_0 = 0$ sec and $t_f = 10$ sec. Figure (8) demonstrates the arm tracking for the optimal path where the red and green links represent the link1 and link2 of the arm respectively.

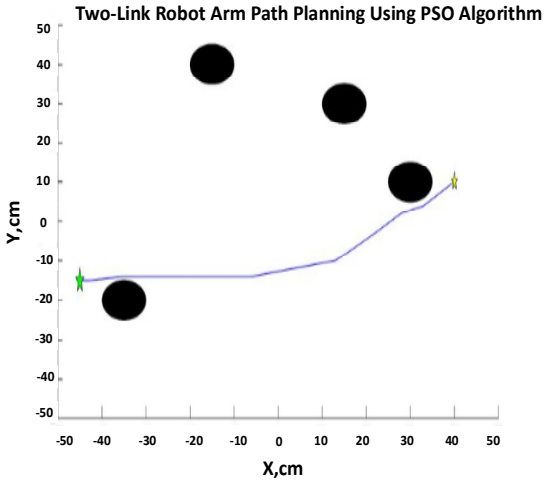


Fig. 7. Quantic polynomial trajectory path optimal solution.

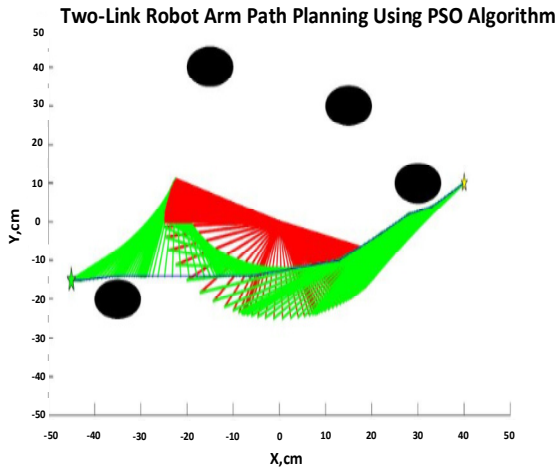


Fig. 8. Two-link robot manipulator optimal path.

The next figures (9, 10, 11, and 12) show the results of changing the x, y, theta 1 of first link and theta 2of second link of the optimal smoothing path during 10s.

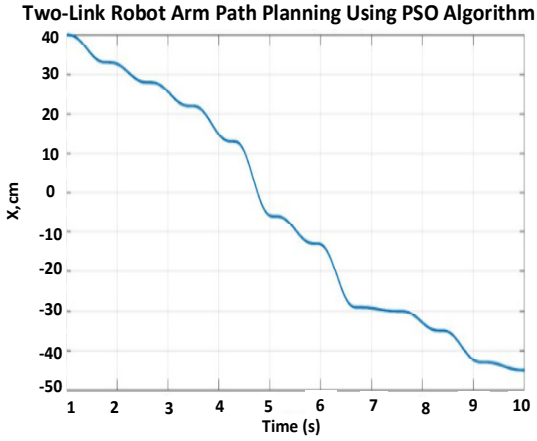


Fig. 9. The change of x coordinate of optimal path.

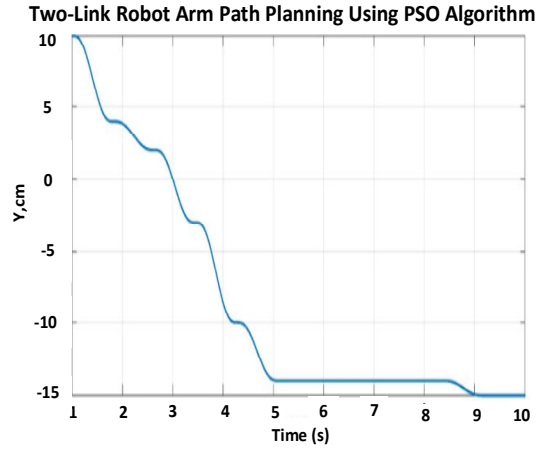


Fig. 10. The change of y coordinate of optimal path.

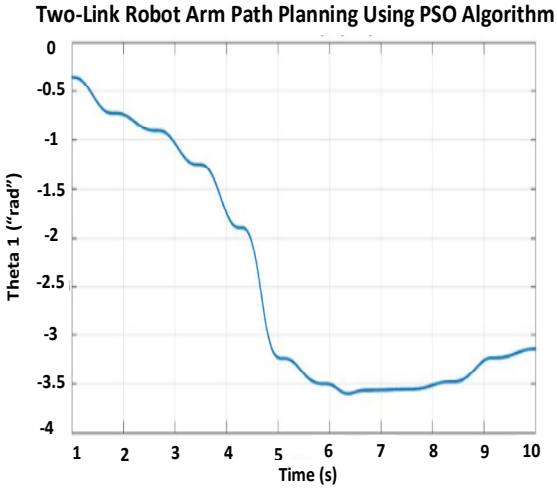


Fig. 11. The change of theta 1 of optimal path.

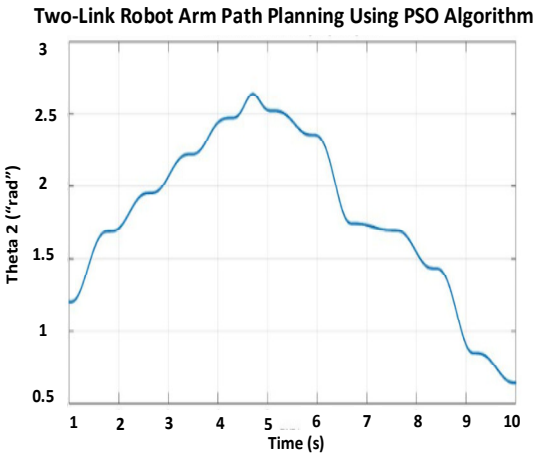


Fig. 12. The change of theta 2 in optimal path.

Another task was tested by the same proposed mapping but different in the start and goal points. The path is from the start point (30, -30) to the goal point (15, 45), where its cost fitness is 82.8733 and executed for 49s, as shown in figure (13, 15), and smoothing path shows in figure (14).

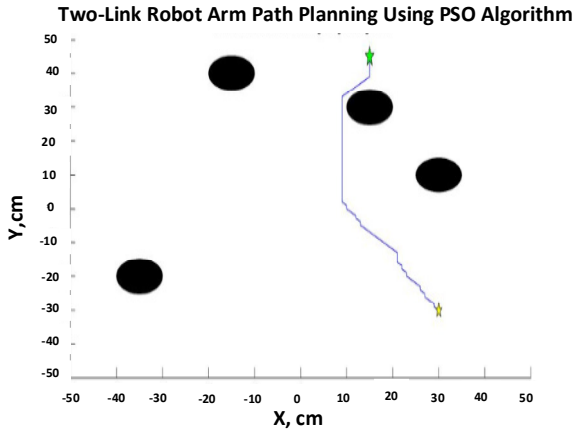


Fig. 13. PSO optimal path solution.

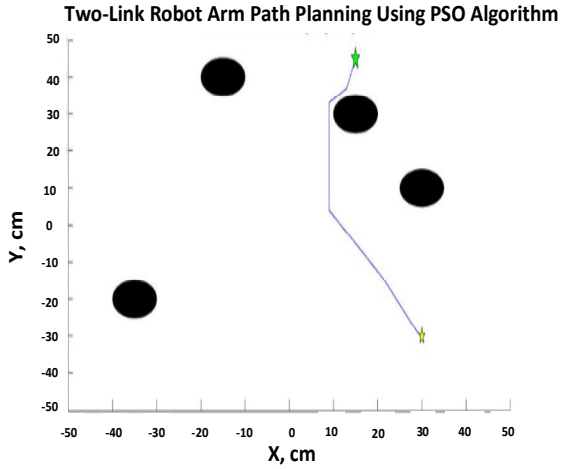


Fig. 14. Trajectory path optimal solution.

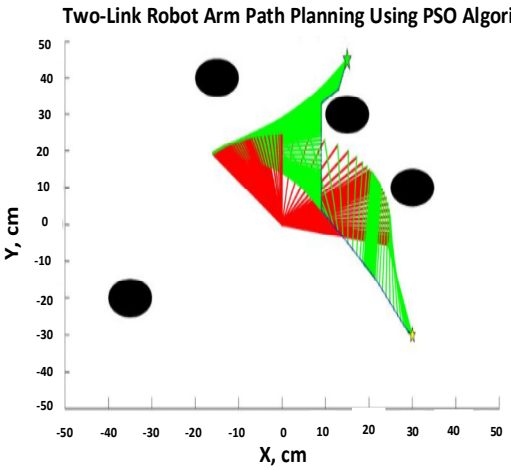


Fig. 15. Two-link robot manipulator optimal path, where the red line denoted the link 1 of arm while the green line denoted the kink2 of arm.

Figures (16, 17, 18, and 19) clarifies the results of changing the x, y, theta 1 of first link and theta 2of second link of the optimal smoothing path during 10s.

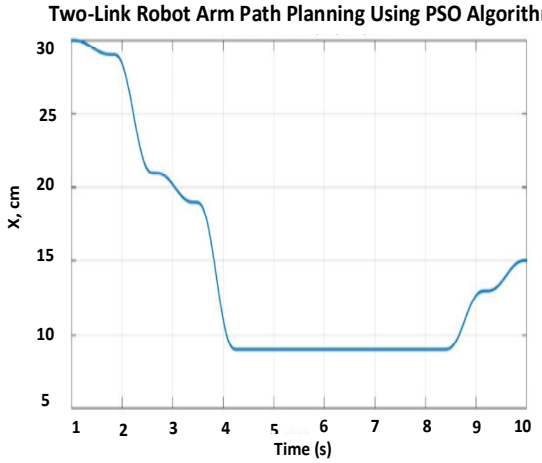


Fig. 16. The change of x coordinate of optimal path.

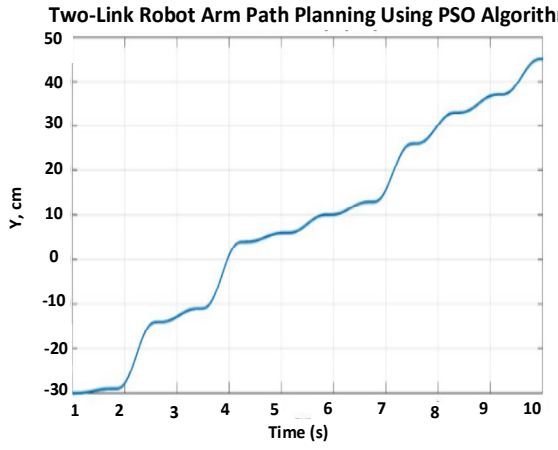


Fig. 17. The change of y coordinate of optimal path.

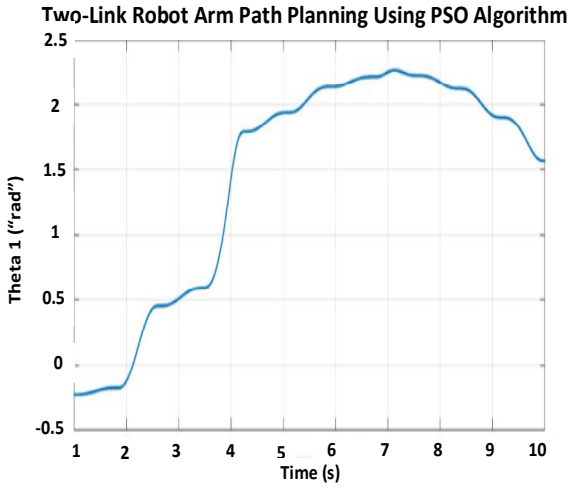


Fig. 18. The change of theta 1 of optimal path.

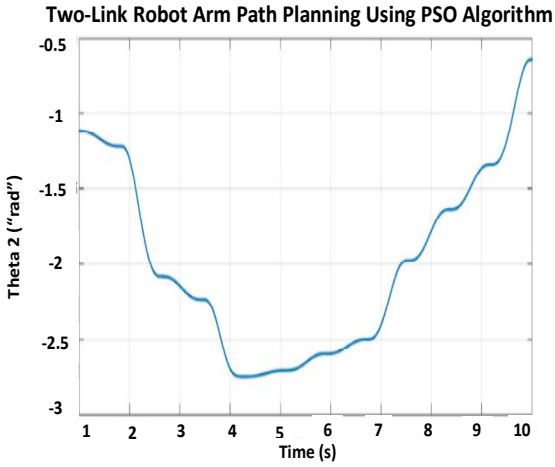


Fig. 19. The change of theta 2 of optimal path.

10. Conclusion

For solving the path planning problem and to find the optimal path of the robot arm, the modified PSO algorithm is proposed. The free Cartesian space with free elbow up solution space and free elbow down solution space are initially constructed by using the inverse kinematics. In this paper PSO is proposed to find the shortest path depending on the costs of D* and Euclidean distance. The modified algorithm efficiently finds the optimal path by locally searching and computing the probability to all possible solutions of robot arm end effector position and takes the best one to form the path. More specifically, (8) and (9) equations are playing an important role in providing high diversity to the possible solutions. Finally, the quintic polynomial equation is integrated with PSO for smoothing the path at specific time.

By comparison with reference [6] that based on CBPSO algorithm to find the near-optimal path for the robot manipulator in the joint space, the results from the table below clearly confirm that the proposed algorithm of modified PSO based on D* algorithm has better performance in any Two-link robot environment. The proposed method has the ability for finding the optimal shortest path solution in many easy and difficult environments.

Table (2) shows test data of computation time (s) generated by CBPSO and D* based PSO algorithms, corresponding to the different number of obstacles existence workspace types of the workspace and optimality of the generated path.

Table 2,
The comparison of different algorithms.

	CBPSO algorithm	**D* based PSO**
Computation Time(s)	64.65	63.5
Number of Obstacles	Less than 4	More than 4
workspace	Joint space	Cartesian space
Optimality of the path	Near to optimal	Optimal shortest path

In contrast, it is possible to observe some ambiguity in the performance of the proposed algorithm in online path planning process with static and dynamic obstacles.

11. Reference

[1] Kelly R., Santibanez V. and Loria A. (2005), "Control of Robot Manipulators in Joint Space", Springer, Heidelberg, New York, Hong Kong, London, Milan, Paris, Tokyo.

[2] Rakibul Islam and Tajmiruzzaman (2014), "Autonomous Robot Path Planning Using Particle Swarm Optimization in Dynamic Environment with Mobile Obstacles & Multiple Target", ICMIEE, Bangladesh.

[3] Ahmed T. Sadiq, A. H. H.(2013),"Swarm Robot Path Planning Using Hybrid PSO with D* Algorithms in Dynamic Environment", Iraq.

[4] ATA, A. A.(2007),"Optimal Trajectory Planning of Manipulators",Journal of Engineering Science and Technology, Vol. 2.

[5] Shahab A., M. Usman Rafique, and M. Umer Khan (2015)," Mobile Robot Path Planning in Static Environments using Particle Swarm Optimization", International Journal of Computer Science and Electronics Engineering (IJCSEE), Vol. 3.

[6] Chao and Liu (2012), " Modelling And Optimization Approach For Trajectory Planning of Three Freedom Planar Manipulators", Journal of Pure and Applied Mathematics: Vol. 7, No.1, pp. 1-20.

[7] Rokbani and Alimi (2013)," Inverse Kinematics Using Particle Swarm Optimization, A Statistical Analysis", Procedia Engineering, Vol. 64.

[8] Kim and Lee (2015)," Trajectory Optimization with Particle Swarm Optimization for Manipulator Motion Planning", IEEE Transaction on Industrial Informatics.

[9] Craig, J. J. (1989)," Introduction to Robotics", Addison-Wesley.

[10] Wesam Mohammed (2011)," Solution of Inverse Kinematics for SCARA Manipulator Using Adaptive Neuro-Fuzzy Network", IJSC, Vol.2, No.4.

[11] Cubero S (2006)," Industrial Robotics Theory Modelling Control", ISBN.

[12] Richard M., Zexiang Li and S. Shankar (1994),"Mathematical Introduction to Robotic Manipulation".

[13] M. O'Malley (2011), "Introduction to Robotics, Inverse Manipulator Kinematics", International Journal on Soft Computing (IJSC) Vol.2, No.4.

[14] W. Spong, Seth Hutchi nson, and M. Vidyasagar (2005), Robot Modeling and Control,1st ed, Wiley.

[15] Carl D. III Crane. By Joseph Duffy (2009), "Kinematic Analysis of Robot Manipulators", Cambridge University Press.

[16] Ahmadzadeh S. and Ghanavati M. (2012)," Navigation of Mobile Robot Using the PSO Particle Swarm Optimization", Journal of Academic and Applied Studies, Vol. 2, pp. 32-38.

[17] S.N.Sivanandam and P.Visalakshi (2007), "Multiprocessor Scheduling Using Hybrid Particle Swarm Optimization with Dynamically Varying Inertia", International Journal of Computer Science & Applications, Vol. 4 Issue 3, pp. 95-106.

[18] Sanjay Sarma O V, Vishwanath Lohit T and Deepak Jayaraj (2012), "Path Planning in Swarm Robots using Particle Swarm Optimization on Potential Fields", International Journal of Computer Applications, Vol. 60.

[19] Xin Chen and Yangmin Li. (2006), "Smooth Path Planning of a Mobile Robot Using Stochastic Particle Swarm Optimization", in Proceedings of the IEEE International Conference on Mechatronics and Automation, pp.1722-1726.

[20] Shi .P and Y. Zhao (2009), "An Efficient Path Planning Algorithm for Mobile Robot Using Improved Potential Field", in IEEE International Conference on Robotics and Biomimetics, pp. 1704-1708.

[21] Nosrati M., Karimi R. and Hasanvand H. (2012)," Investigation of the D * (Star) Search Algorithms: Characteristics, Methods and Approaches", World Applied Programming, Vol. 2, No.4.

تخطيط المسار لذراع الروبوت باستخدام تقنية سرب الطيور المعدلة إعتماداً على خوارزمية *D

احمد طارق صادق* فراس عبد الرزاق رحيم نور الهدى فضل عباس*****

*،*****قسم علوم الحاسوب/ الجامعة التكنولوجية*

******قسم السيطرة والنظم/ الجامعة التكنولوجية*

*البريد الالكتروني: Drahmed_tark@yahoo.com

**البريد الالكتروني: dr.firas7010@yahoo.com

***البريد الالكتروني: Nooralhudafabbas92@yahoo.com

الخلاصة

حظي ذراع الروبوت مؤخرا بالكثير من الاهتمام نظرا لاستخداماته المتعددة في مختلف التطبيقات. ولذلك فان إيجاد المسارالامثل للذراع له دور بالغ الاهمية في تحسين حركات الذراع وتوجيهها . وتتمثل المهمة الاساس لتخطيط المسار في البحث عن المسار الملبي لأهداف الحركة والمتضمنة: تجنب الاصطدام بالعوائق، وخفض زمن الحركة، تقليل تكلفة المسافة وتحقيق القيود الكينماتيكية للذراع. في هذا البحث، تم هيكلة الفضاء الكارتيزي الحر لذراع ثنائي درجات حرية الحركة، والذي يضم جميع قيم المفاصل بضمانة عدم الاصطدام بالعوائق. فضلا عن استخدام كلل من خوارزمية *Dومعادلة المسافة الإقليدية للحصول على المسافة الدقيقة والمسافة المقدرة عن الهدف على التوالي. واقتراح خوارزمية السرب المعدلة للبحث عن المسار الامثل محليا اعتمادا على المسافات المستخلصة من خوارزمية *D ومعادلة المسافة الإقليدية. وقد تم استخدام المعادلة المتعددة الحدود من الدرجة الخامسة بغية الحصول على مسار ناعم في مدة زمنه محددة. ووفقا للنتائج المستخلصة، فأن خوارزمية السرب المعدلة تبحث بكفاءة عن المسار الأمثل وفي ظل بيئات صعبة ومعقدة.