Al-Khwarizmi Engineering Journal,Vol. 13, No. 4, P.P. 124- 136 (2017)

Al-Khwarizmi Engineering Journal

Applying A* Path Planning Algorithm Based on Modified C-Space Analysis

Firas A. Raheem* **Asmaa A. Hussain****

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

*Email: 60124@uotechnology.edu.iq

**Email: asmaa.alshawi@yahoo.com

(Received 25 October 2016; accepted 7 March 2017)

https://doi.org/10.22153/kej.2017.03.007

Abstract

In this paper, a modified derivation has been introduced to analyze the construction of C-space. The profit from using C-space is to make the process of path planning more safety and easer. After getting the C-space construction and map for two-link planar robot arm, which include all the possible situations of collision between robot parts and obstacle(s), the A* algorithm, which is usually used to find a heuristic path on Cartesian W-space, has been used to find a heuristic path on C-space map. Several modifications are needed to apply the methodology for a manipulator with degrees of freedom more than two. The results of C-space map, which are derived by the modified analysis, prove the accuracy of the overall C-space mapping and construction, and then a successful and guaranteed path from a start to goal configuration has been obtained without any collision probability. The results had been achieved by (Matlab R2015a) software, which run on Intel (R) Core (TM) i3-3120M CPU.

Keywords: *Configuration Space, Degrees of Freedom, Topology, Path Planning.*

1. Introduction

Due to the importance of path planning and obstacle avoidance problems in practical areas, there are more researches and studies about it in the last decades. In the classical mechanics, the configuration space (C-space) represent very important tool to describe and analyze the motion planning for many important systems, especially in robotics [1]. To understand the C-space, first it is important to know what the term of configuration means in robotic systems. It means a complete description, which determines every point position on the robot uniquely. So, the C-space of a robot typifies the space of all configurations, which are processed by the robot. As the robot has *n* degrees of freedom, the C-space for this robot is *n*-dimensional [2].

C-space deals with static obstacles, or dynamic obstacles with slow motion. Also, the shape and dimensions of each obstacle must be known before finding C-space construction if the path planning that will be used is off-line.

From the first time the C-space was defined [3], it played very important role in robotic systems especially in path finding problem. Then, the usage of C-space continue in many researches and applications. Until today, more analysis and modifications with many type of robots are applied on C-space [4- 7].

Path planning for robot arm requires optimized global path generation, in which the robot arm can avoid any obstacle in the Cartesian W-space. The feasibility of path planning depends on some factors: map accuracy, number of obstacles and current configuration of the robot. There are many approaches and algorithms have been proposed to solve the path planning problem. This problem is related with finding the shortest path between the start and goal configurations. The most known algorithms that are being used to find shortest path are graph search algorithms. These algorithms are based on node-edge marking [8]. Graph search methods speculate the cost and generate a sequence of segments that extent from the start point to the goal point through vertices [9]. One of the graphical search methods is the A* algorithm. It is a graph, heuristic and greedy search algorithm, that means the determined solution is approximated not optimal (it is Best-First Search) [10]. The main advantage of A* that it can be modified easily and by change the weights of the traversal costs, it can be extended [11]. Usually A* algorithm is applied to find the path on Cartesian W-space, while in this paper, a different approach adopted by the application of A* algorithm on C-space analysis, which includes a modified derivation based on geometrical analysis to find the construction of C-space for two revolute joints planar robot arm.

2. Modified C-Space Derivation

For a manipulator, the space that represented by the variable parameters of joints, either slide position or rotation angles, is called C-space. It includes two main regions: C-free and C-obstacle, in which the second region is a prohibitive because it represents collision between the robot arm and obstacles and between the links of the robot arm [4]. The geometrical representation of the obstacle (s) can be denoted as B, so the configuration space obstacles will be as below [1]:

$$C_{obs} = \{q: A(q) \cap B \neq \emptyset\} \quad \dots (1)$$

While C-free can be represented as below [1]:

$$C_{free} = \{q: A(q) \cap B = \emptyset\} \quad \dots (2)$$

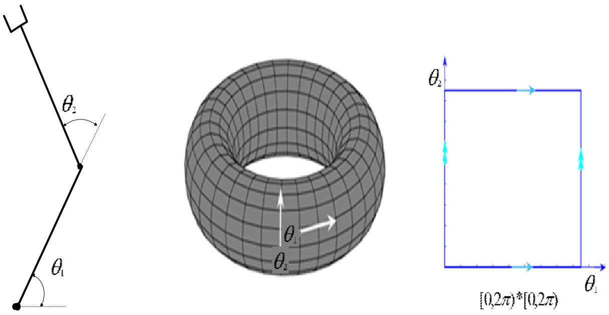
Where:
C_{obs} : C-space obstacles
q: Robot configuration
$A(q)$: The set of points of the space confined by the robot at configuration q
The C-space as overall can be defined as [1]:

$$C_{space} = C_{obs} \cup C_{free} \quad \dots (3)$$

The mathematical study of the surface properties, which are do not change if this surface has been applied for certain deformations, such as bending or stretching, is called topology of space. The topology for robot arm with two revolute joints can be described as [12]:

$$S^1 * S^1 = T^2 \quad \dots (4)$$

The above equation means that the C-space for two revolute joints robot arm is a torus (with no joint limits for the joints). Where S^1 is the circle description, and $T^2$is the 2-dimensional torus surface [12]. Each joint angles pair is corresponded by a unique point on the torus.



(a) 2R Robot Arm (b)2Torus (c)Sample representation

Fig. 1. Topologically representation of the 2-dimensional C-space [12].

To find the mapping of C-space, there are some basic shapes of the obstacle(s) will be analyzed.

2.1. C-Space Construction for Point Obstacle

Mathematically, the simplest obstacle may be found in the workspace is the point. In this paper, a modified derivation for C-space analysis will be discussed, in which the collision checking based on geometrical analysis.

1) First Link-Point Obstacle

A collision between the first link and the point obstacle will acquire if the following condition achieve:

$$\theta_{obs} = \theta_1 \quad and \quad x_1 \geq m \quad and \quad y_1 \geq n \quad \dots (5)$$

Where:

$$\mathrm{d}_{obs} = \frac{n-0}{m-0} \ ; \qquad \theta_{obs} = \tan^{-1}\frac{n}{m} \ ; \qquad \dots (6)$$

x_1, y_1: First link end effector coordinate
m, n: Point obstacle coordinate

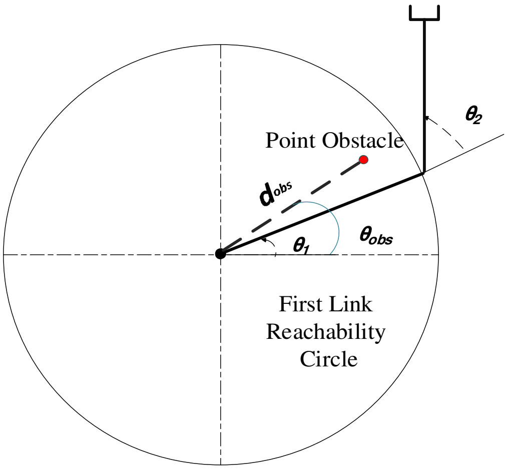


Fig. 2. The first link with the point obstacle geometrical analysis.

For Fig. 2, the coordinate of point obstacle is inside the first link reachability circle. If the slope of the first link equal to the slope of the point obstacle, the collision is acquire.

2. Second Link-Point Obstacle

The following mathematical derivation explains the condition of collision between the second link and point obstacle:

$$\alpha = \pi - \theta_2 \quad \text{... (7)}$$

$$\gamma = \theta_{obs} - \theta_1 \quad \text{... (8)}$$

By sine's law:

$$\frac{L_1}{\sin\beta} = \frac{\mathrm{d}_{obs}}{\sin\alpha} \Longrightarrow \beta = \sin^{-1}\frac{L_1 * \sin\alpha}{\mathrm{d}_{obs}} \quad \text{...(9)}$$

While,

$$(\alpha + \gamma + \beta) = \pi \quad \text{... (10)}$$

There is a collision between the robot and the point obstacle, where (α, γ and β) are the angles used for the analysis as shown in Fig. 3.

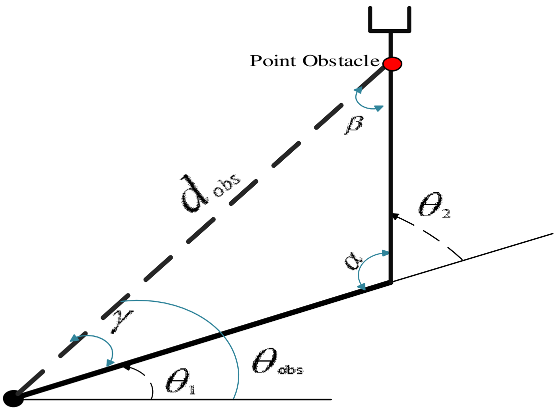


Fig. 3. The second link with the point obstacle geometrical analysis.

These collision conditions are applied for any position of the point obstacle inside first or second link, not only at the end effector for each of them.

2.2. C-Space Construction for Line Obstacle

The robot links can be handled as two lines, with regardless of their width. In our research, the modified derivation of C-space analysis includes two analyses for checking collision.

Analysis One

To implement the checking, there are several constraints can be put:

1) First Link – Line Collision

In Fig. 4, if the first joint angle, at specific configurations, has values locate at the range of angles of end points for line obstacle, this may cause collision between the first link and the obstacle:

$$\theta_{l1} \leq \theta_1 \leq \theta_{l2} \quad \text{... (11)}$$

Where:

θ_{li} : The angles of line obstacle end points, i = 1, 2.

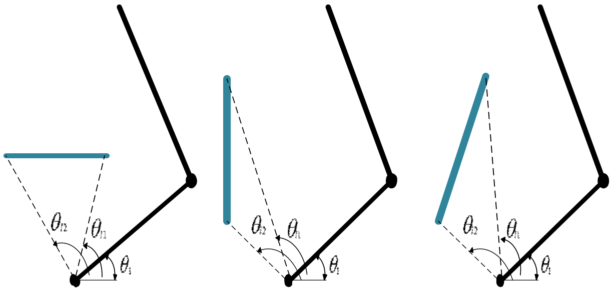
It is important to take in consideration that the length of the first link equal or more than the distance between the robot's base and the line obstacle, as in the following equation:

$$[x_1, y_1] \geq [x_{Lj}, y_{Lj}] \quad \text{... (12)}$$

$[x_1, y_{1]}$: The coordinate of first link end effector

$[x_{Lj}, y_{Lj}]$: Matrix of line obstacle points

j=1, 2, …, number of line obstacle points

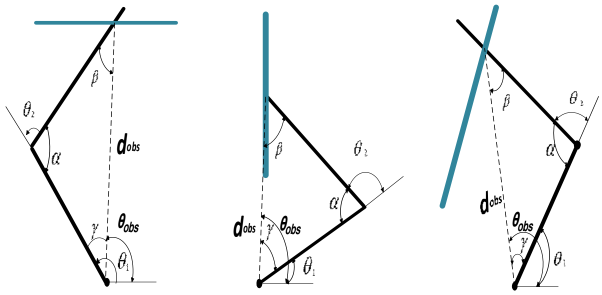


(a) Horizontal Line (b) Vertical Line (c) Sloping Line
Fig. 4. Cases of first link – line obstacle collision.

1) Second Link – Line Collision

As it was done in checking collision with point obstacle, the geometrical analysis can be used to find the collision angle between the second link and the line obstacle. Equations (5, 6, 7, 8, and 9) can be used but it is applied for the intersection point between the second link and the line obstacle.

Fig. 5 illustrate the situations of collision between the second link and the line obstacle.



(a) Horizontal Line (b) Vertical Line (c) Sloping Line
Fig. 5. Cases of second link – line obstacle collision.

Analysis Two

Another method for checking collision by checking the equality of each link slop with the slop of each point in the line obstacle:

For the first link:

$$\frac{y_1-0}{x_1-0}=\frac{y_{Lj}-0}{x_{Lj}-0} \quad \text{... (13)}$$

For the second link:

$$\frac{y_2-y_1}{x_2-x_1}=\frac{y_{Lj}-y_1}{x_{Lj}-x_1} \quad \text{... (14)}$$

An important condition must be considered for equations (13) and (14): the length of each link should be more than or equal to the distance of the robot's base to that point (for equation 13) or from the first link end effector to the point (for equation 14).

2.3. C-Space Construction for Polygonal Shapes Obstacle

Any two dimensions shape formed by straight lines, is called polygon. It has three lines (such as triangle), four lines (such as square, rectangle, rhombic… etc.), or more [13]. So the same methods that have been used for line obstacle can be used here.

2.4. C-Space Construction for Circle Obstacle

For a circle that is known in center (x_c, y_c) and radius (*rad*), the collision can be checked by the following modified derivation of C-space analysis:

The distance from the center of the circle to each point on the first and second link will be calculated:

For the first link:

$$d_1(s)=\sqrt{(x_c-s*x_1)^2+(y_c-s*y_1)^2} \quad \text{... (15)}$$

For the second link:

$$d_2(s)=\sqrt{(x_c-X_2)^2+(y_c-Y_2)^2} \quad \text{... (16)}$$

$$X_2=x_1+s*L_2*\cos(\theta_1+\theta_2) \quad \text{... (17)}$$
$$Y_2=y_1+s*L_2*\sin(\theta_1+\theta_2) \quad \text{... (18)}$$
$$D_1=\min[d_1(s)] \quad \text{... (19)}$$
$$D_2=\min[d_2(s)] \quad \text{... (20)}$$

Where:

$s=0.1, 0.2, \dots, S$

It is important to denote that s is the resolution step of deviation of each link, while S is the total length of link. Theoretically, the step s may be taken 0.1, smaller than 0.1, or biggest. If the following condition verified, that means there is a collision between the circle obstacle and the robot arm:

$$D_i \le rad \quad \text{... (21)}$$

$i=1,2.$

The distances d_i can be implemented by the form of deviation as below:

$$d_{ip}=[d_{i1}, d_{i2}, \dots, d_{iP}] \quad \text{... (22)}$$

P: Number of steps of deviation (s).

Fig .6 show the distances that used in the analysis for checking collision with circle obstacle.

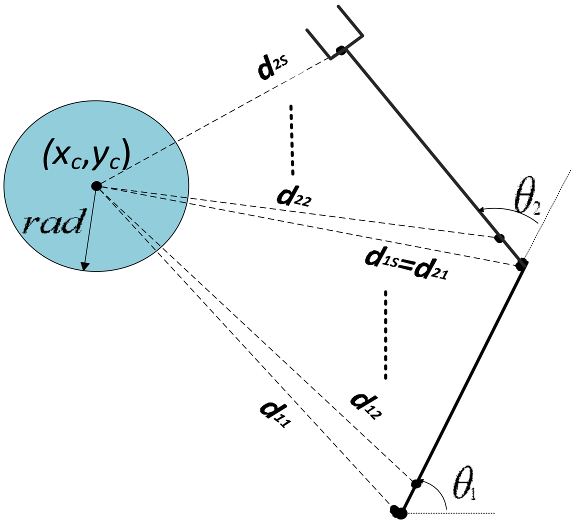


Fig. 6. Geometrical analysis of circle obstacle.

3. A* Algorithm

A* algorithm is most familiar search algorithm that is used to find the shortest path. It represent an extension of Dijkstra's algorithm. The usage of heuristic is what make the A* different from another graph search algorithms. The heuristic provides an estimated distance from a current node to the goal node, it denoted by $h(n)$, also the A* take in account the cost from the start node to the current node, and it is denoted as $g(n)$. The cost function is determined as below [9]:

$$f(n)=g(n)+h(n) \quad \text{... (23)}$$

For C-space map, each node represented by a pair of joint angles, *n:* (θ_1, θ_2).

The A* algorithm includes two lists: OPEN list (O_List) and CLOSED list (C_List). It choose the nodes that have minimum cost functions to create a path from start node to goal node. The fulfillment of this algorithm on C-space map can be explained briefly by the following flowchart:

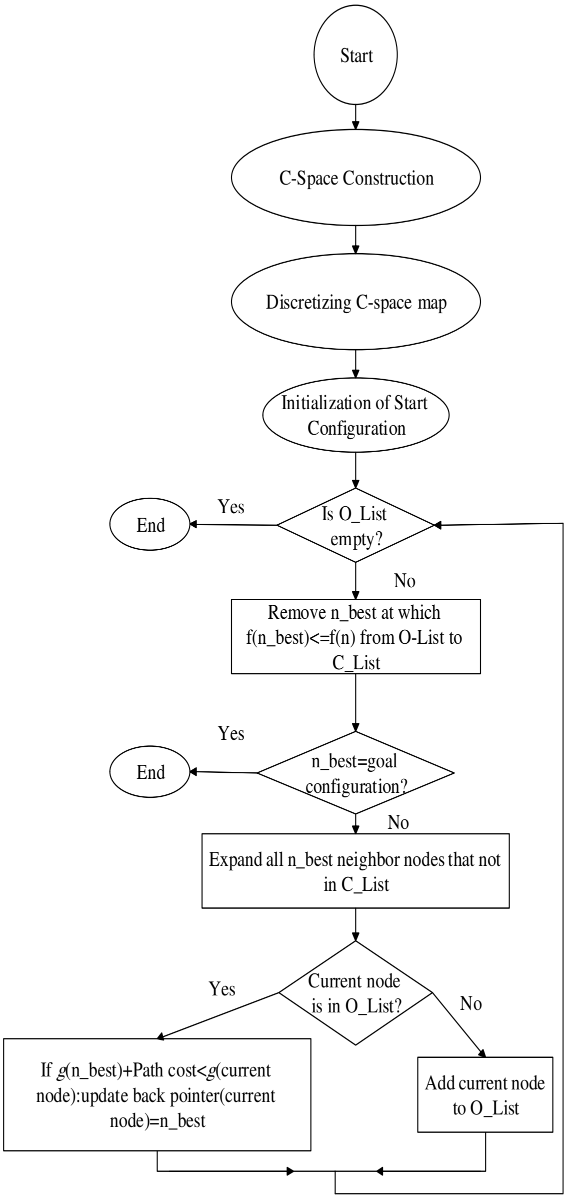

Start
C-Space Construction
Discretizing C-space map
Initialization of Start Configuration
Is O_List empty?
Yes
End
No
Remove n_best at which f(n_best)<=f(n) from O-List to C_List
Yes
End
n_best=goal configuration?
No
Expand all n_best neighbor nodes that not in C_List
Current node is in O_List?
Yes
No
If g(n_best)+Path cost<g(current node):update back pointer(current node)=n_best
Add current node to O_List

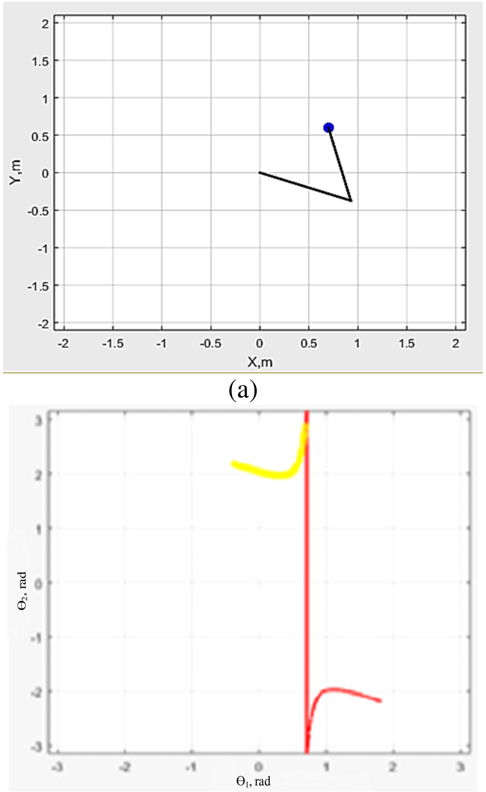
Fig. 7. A* algorithm on C-space map flowchart.

4. Results and Discussion

The results of finding C-space construction and applying A* algorithm on it, will be displayed and discussed in section 4.1 and 4.2 respectively.

4.1. C-Space Construction Results

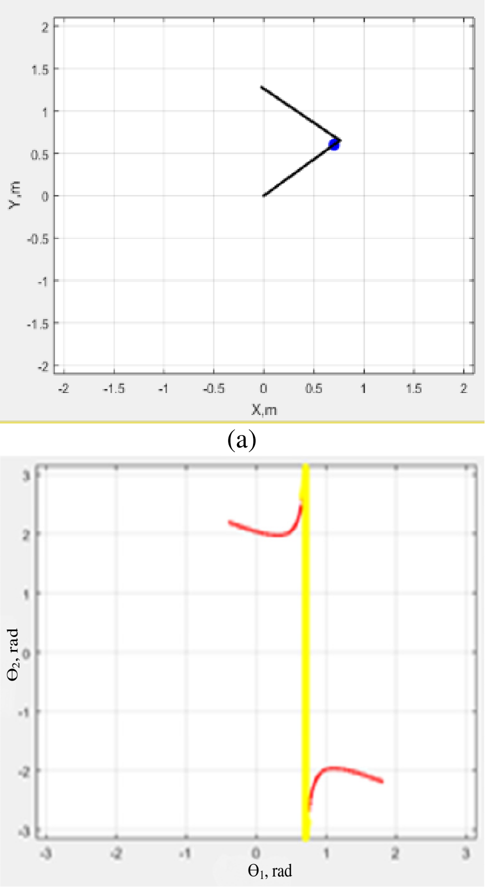
For the point obstacle, it can be observed that the C-space map of the collision between the first link and the point obstacle has the straight-line shape, Fig. 9, while the collision between the second link and the point obstacle has a curved shape. The upper curvature is the result of elbow down configuration, Fig. 8, and elbow up configuration form the lower curvature as in Fig. 10.



(a)

(b)

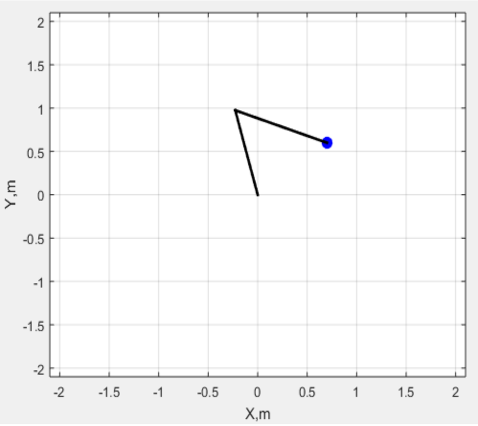
Fig. 8. (a) Second link-point collision in elbow down in Cartesian W-space. (b) Corresponding C-space map.



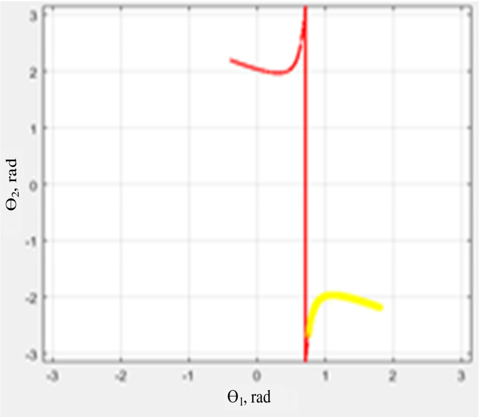
(a)

(b)

Fig. 9. (a) First link-point collision. (b) Corresponding C-space map.



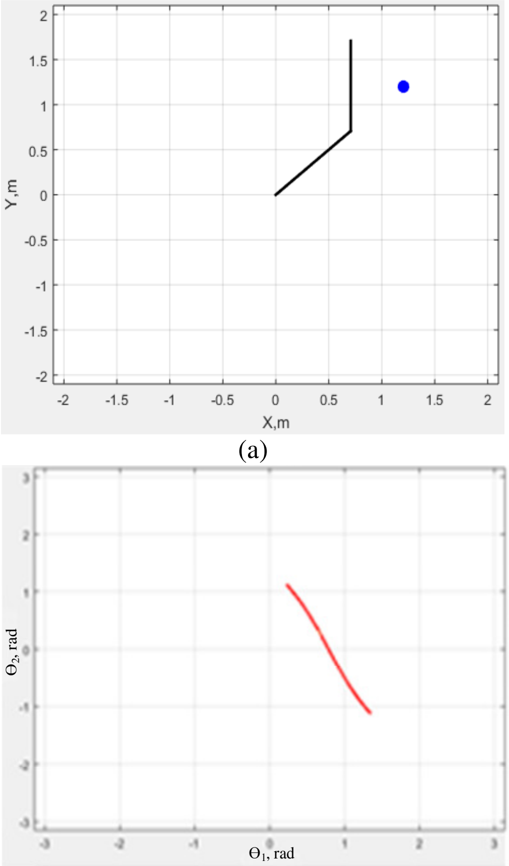
(a)



(b)

Fig. 10. (a) Second link-point collision in elbow up. (b) Corresponding C-space map.

As the point obstacle approaching to the end effector of the robot arm, the C-space obstacle points will decrease, as shown in Fig. 11.

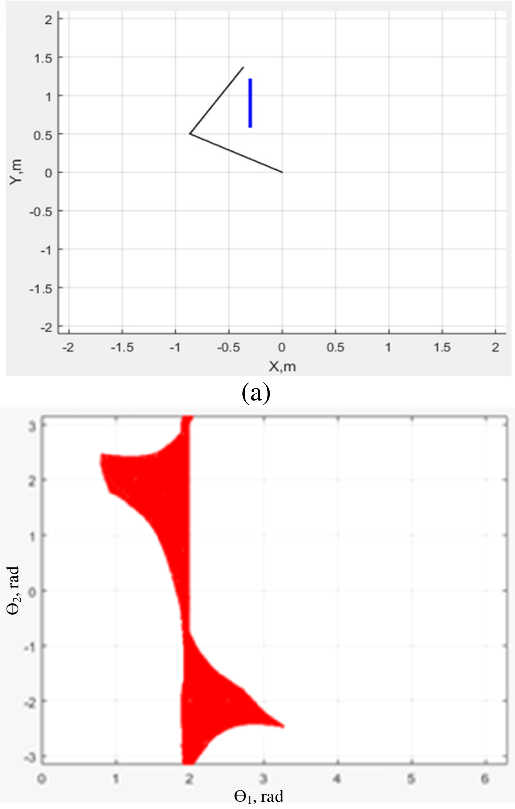


(a)

(b)

Fig. 11. (a) Point obstacle collides with second link only. (b) Curve points decrease as the point be nearest to the end effector.

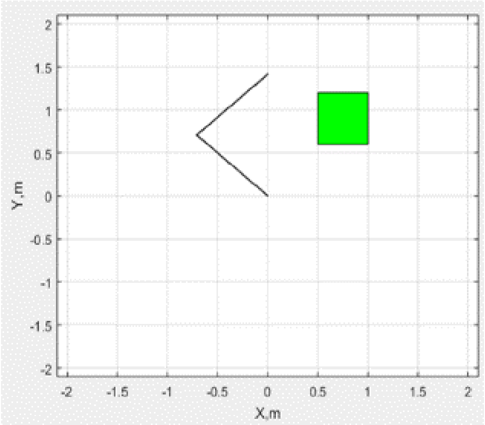
The C-space map of line obstacle and polygonal shape obstacle is achieved by the analysis introduced previously in section 2.2 and 2.3. The same shape of the curve of C-space map for point obstacle applied for each point on the line(s). Here, these curves in compact form will represent the C-space line or polygonal obstacles, as illustrated in Fig. 12 through Fig.15.



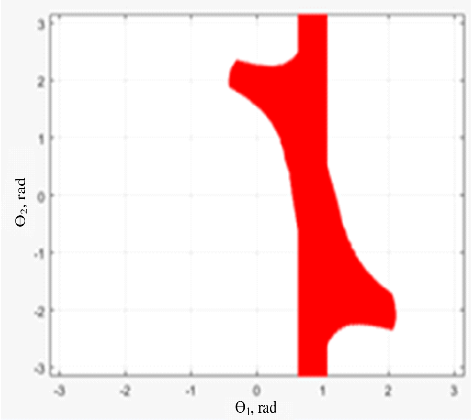
(a)

(b)

Fig. 12. (a) Robot arm with line obstacle in workspace. (b) Corresponding C-space map.

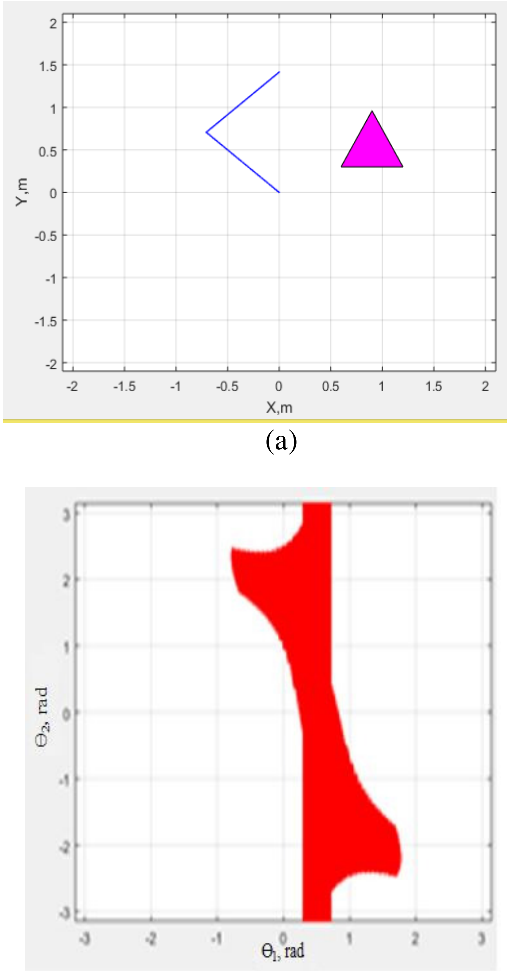


(a)



(b)

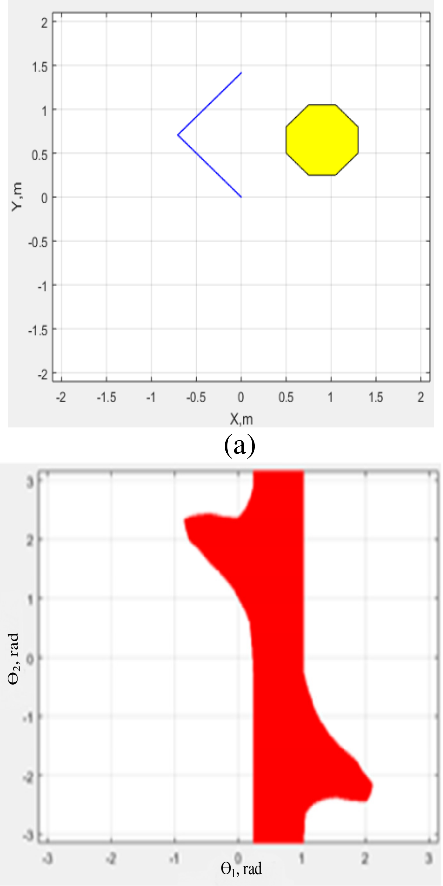
Fig. 13. (a) Robot arm with square obstacle in Cartesian W-space. (b) Corresponding C-space map.



(a)

(b)

Fig. 14. (a) Robot arm with rectangular obstacle in Cartesian W-space. (b) Corresponding C-space map.



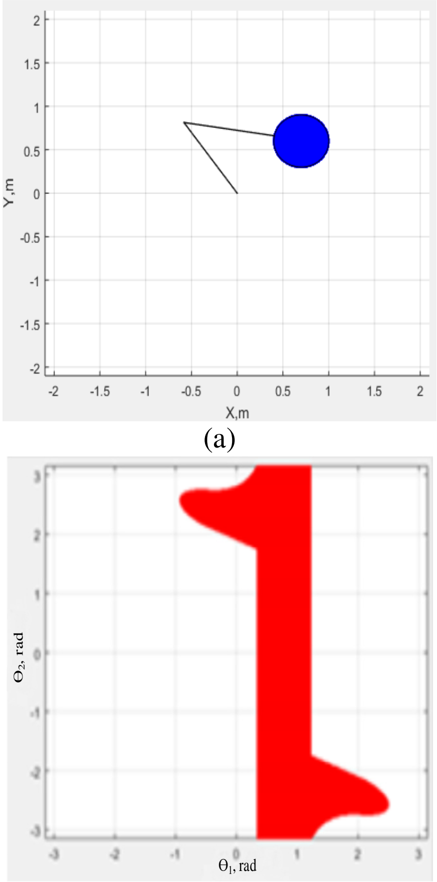
(a)

(b)

Fig. 15. (a) Robot arm with octagonal obstacle in Cartesian W-space. (b) Corresponding C-space map.

The figures (12-15), represent the case of collision with both links. The number of points on C-space map are increase as the obstacle in the Cartesian W-space be near to the first link. In addition, it can be observed that the C_{free} is on the both sides of the C_{obs}. Therefore, to find the path from one side to other, the range of changing θ_1 is chosen in somehow for unification the region of C_{free} which make the path planning easier.

For circle obstacle**,** Fig. 16, there are two main parts of edges of the C-space map. First, the collision between the first link and the circle obstacle circumference, which represented by the vertical straight lines with all values of θ_2. Second, the outer upper and lower curved shapes represents the collision between the second link and the circle obstacle circumference.



(a)

(b)

Fig. 16. (a) Robot arm with circular obstacle in Cartesian W-space. (b) Corresponding C-space map.

For any C-space obstacle shape, the map of C-space is formed by matched angles. Each pair of matched points in the obstacle in Cartesian W-space has a pair of matched angles in C-space. For example:

Table 1,
Two Matched Points in Cartesian W-Space and Their Symmetry Angles in C-space

X(m)	Y(m)	θ_1(deg.)	θ_2(deg.)
0.5808	0.2111	-52	144
0.211	0.5808	142	-144

This pair of points represent the first point and last point on the C-space map for circle obstacle with center of (0.5, 0.5) and radius equal to 0.3 m, which is illustrated in Fig. 17. It can be seen that the symmetry points on C-space have the same θ_2 value with different sign, which is verifies the two solutions of elbow up and elbow down of the second joint.

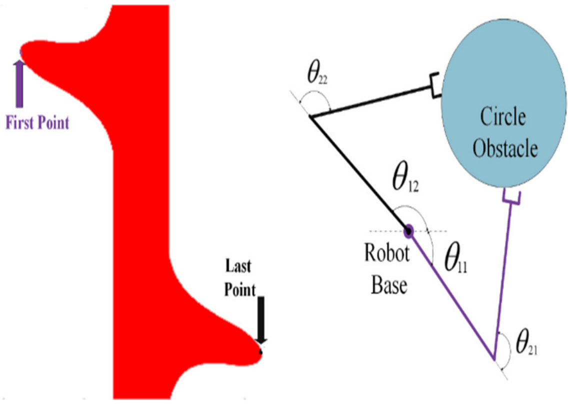


Fig. 17. The matched points analyzing for circle obstacle.

For any obstacle shape, if there is a collision between the robot links and the circumference of that obstacle, by the way any inner point of the obstacle has a corresponding point on the C-space map.

4.2. Results of Applying A* Algorithm on Modified C-Space

The following two environments have different obstacle shapes and tasks. First, we do all the steps of C-space analysis and then the construction to get the exact map, which includes the collision and free areas. Second, the start and goal configuration, which are represented by the dot and the star shape respectively, have been computed by the inverse kinematic equations.

For the first and second environment, the start and goal points represented by the dot and the star shape respectively.

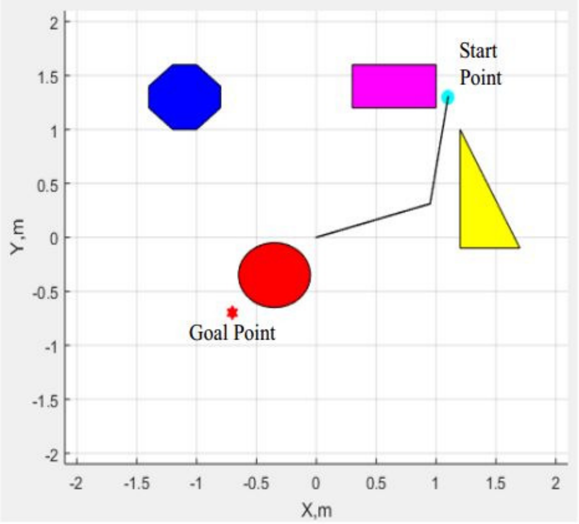


Fig. 18. Cartesian W-space of the first environment.

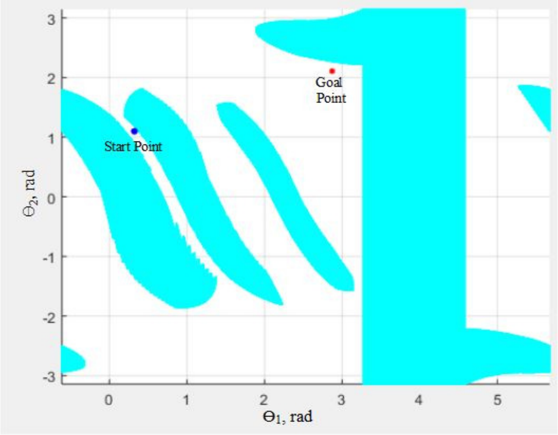


Fig. 19. C-space map of the first environment.

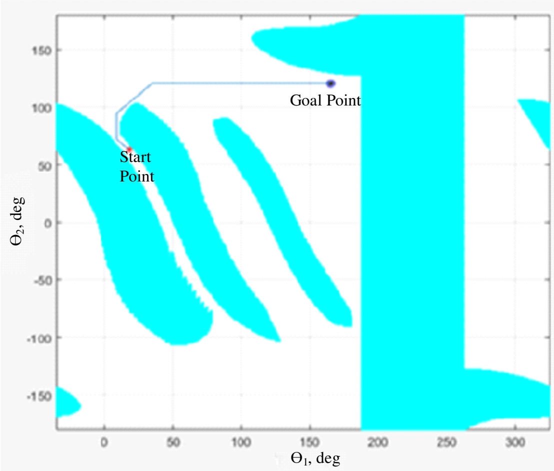


Fig. 20. A* path in C-space map for the first environment.

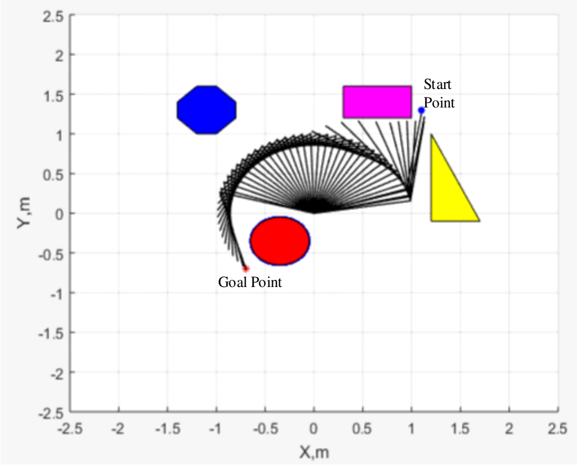


Fig. 21. The movement of robot arm from start to goal point for the first environment.

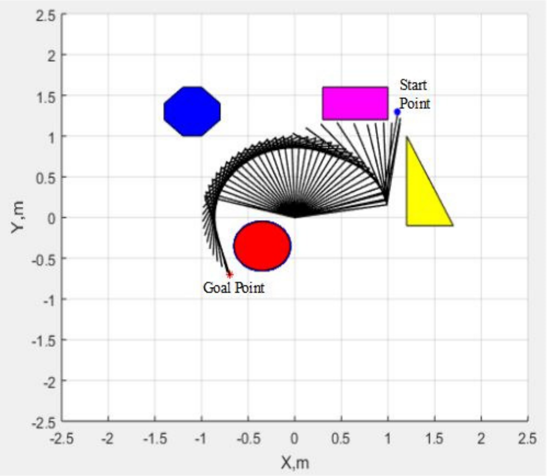


Fig. 22. Two-link robot arm end effector path in Cartesian W-space for the first environment.

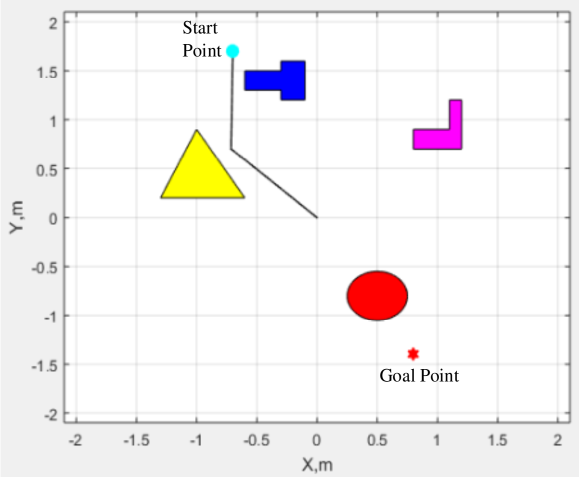


Fig. 23. Cartesian W-space of the second environment.

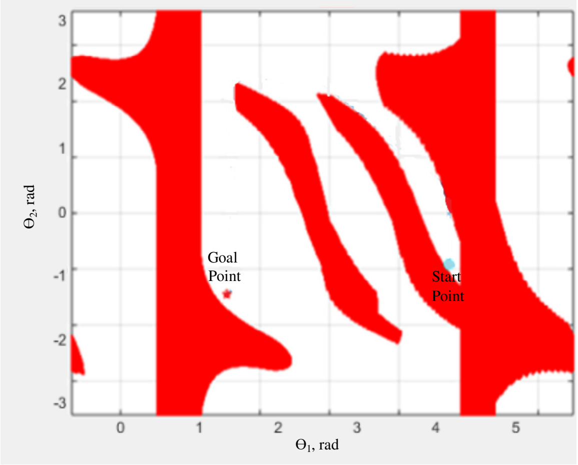


Fig. 24. C-space map of the second environment.

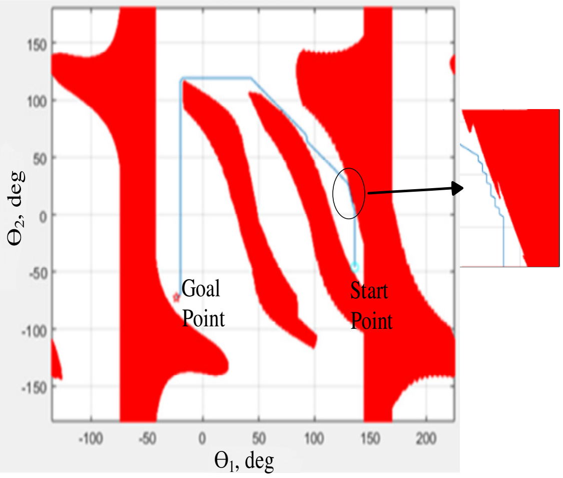


Fig. 25. A* path in C-space map for the second environment.

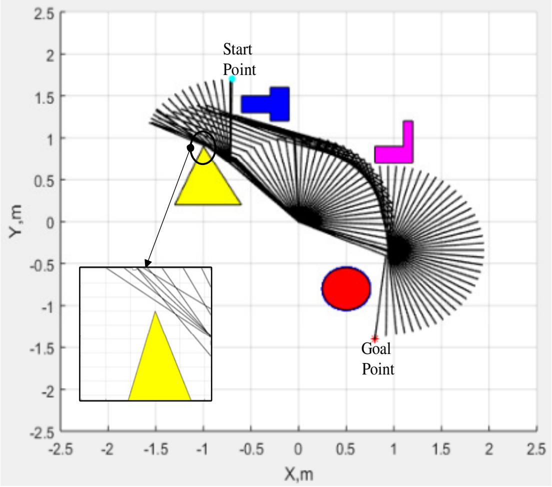


Fig. 26. The movement of robot arm from start to goal point for the second environment.

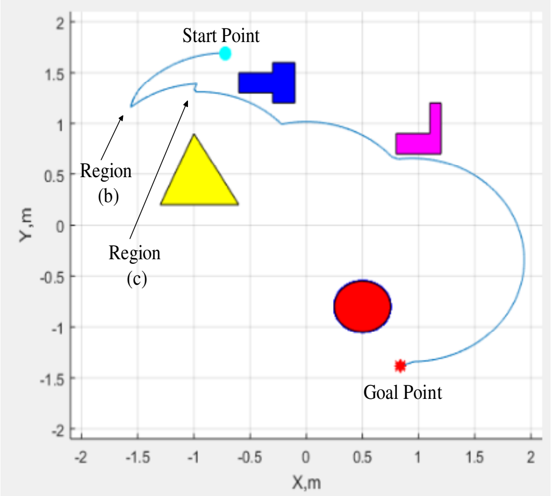


Fig. 27. Two-link robot arm end effector path in Cartesian W-space for the second environment.

From Fig.18 and Fig.23, both environments considered as a crowded environments because all obstacles are closed to the arm links with different distances. After getting the C-space construction of these environments, Fig.19 and Fig.24 show the overall C-space map, which include the C_{free}, C_{obs} and the start and goal configuration. According to the A* path on C-space map, as observed in Fig.20 and Fig.25, the movement of the robot arm prove the success of A* algorithm on these maps, because in spite of the nearness of obstacles from the robot links, it can rescue itself from the crowded region and complete a smoothed path, as in Fig.21 and Fig.26. In addition, it is clear that if the A* path on C-space is optimal; it is not a condition to be optimal on Cartesian W-space for the same environment, and vice versa, but it represent a best solution for planning the path of the two revolute joints planar robot arm between obstacles. In the end effector path for both environments, Fig.22 and Fig.27, it is transparent there are several curved regions, denoted by region (a) in Fig.22 and regions (b) and (c) in Fig.27, represent a reflection of robot arm carful movement at these regions to avoid collision with obstacles. This sudden change insure the complete avoidance of the robot arm from obstacles.

The resulted path and the changing of joint angles in C-space has the behavior shown in Fig. 28 and Fig. 29 respectively:

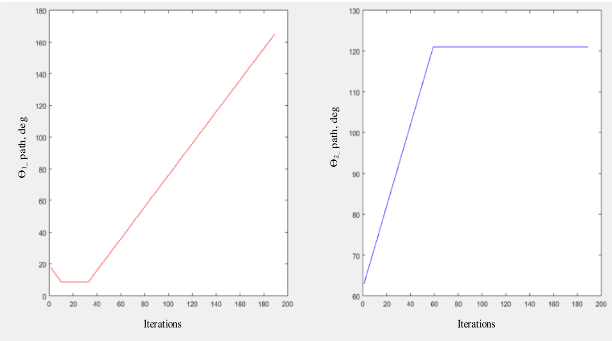


Fig. 28. The path of first and second joint angles along the iterations for the first environment.

The change of joint angles can be calculated from the following equation:

$\Delta\theta_i = \theta_i(t) - \theta_i(t-1)$ ... (24)

Where:

$\Delta\theta_i$= Change of joint angles, i= 1, 2.

t: Current iteration.

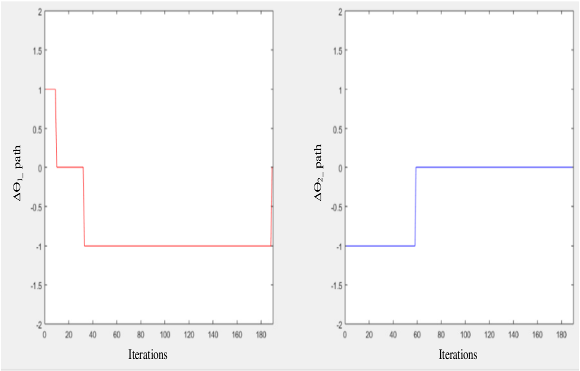


Fig. 29. Changing of first and second joint angles along iterations for the first environment.

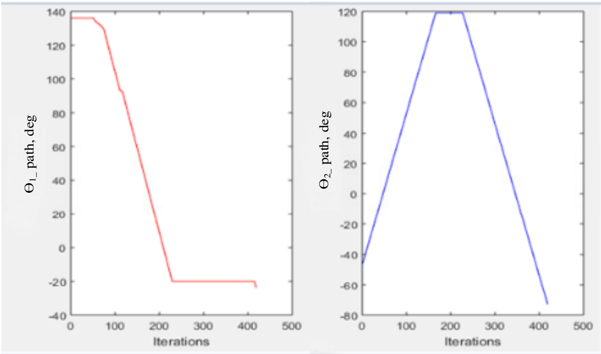


Fig. 30. The path of first and second joint angles along the iterations for the second environment.

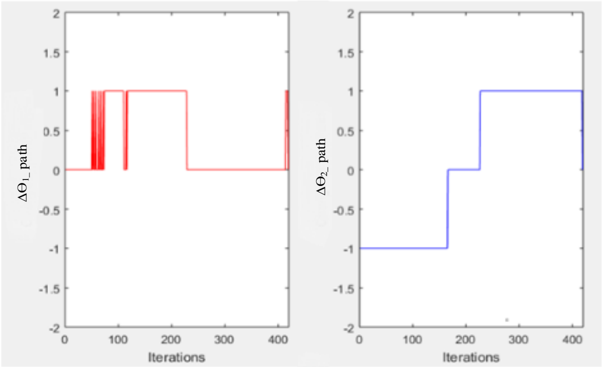


Fig. 31. Changing of first and second joint angles along iterations for the second environment.

From the above results, Fig. 30 and Fig. 31, the path and the behavior of changing each joint angle for the first environment is smooth. while the results of the second environment show there is a little of zigzag in the behavior of first joint angle because of the nearness of the triangular obstacle from the end effector path, so that was reflected on C-space by making the path take the form of the same obstacle border at particular region.

5. Conclusion and Discussion

The methodology of this work can be applied for any type of robot manipulator. Here, we tested it on 2R planar robot arm. In C-space, in general, the robot arm at specific configuration can be represented as a point in C-space map, which represent the values of first and second joint angles at this configuration. The step of changing joint angles had been chosen with particular value in order to achieve sufficient discretization resolution for getting accurate C-space map. The modification on C-space has been done by checking all the odds of collision by modified derivation depending on geometrical analysis. This analysis ease in application and gave accurate and exact C-space map. After applying A* algorithm on modified C-space, we achieve a shortest, safe, and rather smooth path. (Matlab R2015a) was used to program the C-space construction and the algorithm of A*, that was by Intel (R) Core (TM) i3-3120M CPU. The key advantage of applying A* on C-space map not on Cartesian W-space is insuring the path in the safe area. For future work, the A* algorithm results can be enhanced by using classical interpolation equations.

6. References

[1] Jia Pan, Dinesh Mancha, "Efficient Configuration Space Construction and Optimization for Motion Planning", Published by Engineering Sciences Press, 2015.

[2] Benjamin Wah, "Robot Motion Planning", Wiley Encyclopedia of Computer Science and Engineering, 2008.

[3] T. Lozano- Pérez, "Spatial Planning: A Configuration Space Approach", IEEE Computer Society Washington, DC, USA, 1983.

[4] Kasper Althoefer, "Neuro-Fuzzy Motion Planning for Robotic Manipulators", Department of Electronic and Electrical Engineering, King's College London, University of London, 1996.

[5] Wyatt S. Newman, Michael S. Branicky, "Real-Time Configuration Space Transforms for Obstacle Avoidance", The International Journal of Robotics, 1991.

[6] Debanik Roy, " Serial and Parallel Robot Manipulators – Kinematics, Dynamics, Control and Optimization ", Edited by Serdar Küçük, Janeza Trdine 9, 51000 Rijeka, Croatia, 2012.

[7] [Yan Yan, "Geometric Motion Planning Methods for Robotics and Biological Crystallography", Baltimore, Maryland, 2014.

[8] S. M. Masudur Rahman Al-Arif, A. H. M. Iftekharul Ferdous, Sohrab Hassan Nijami, " Comparative Study of Different Path Planning Algorithms: A Water based Rescue System ", International Journal of Computer Applications (0975 – 8887) Volume 39– No.5, February 2012.

[9] Sung Hyun Kim, "Multi-Layer Approach to Motion Planning in Obstacle Rich Environment", Texas A&M University, May 2008.

[10] Suraya Binti Abu Bakar, "High-Speed Shortest Path Co-Processor Design", University of Malaya, Kuala Lumpur, October 2010.

[11] Steven Bell Student Member, "An Overview of Optimal Graph Search Algorithms for Robot Path Planning in Dynamic or Uncertain Environments", IEEE Student Member No. 90691022, Oklahoma Christian University, 19 March 2010.

[12] F. C. Park, K. M. Lynch, "Introduction to Robotics: Mechanics, Planning, and Control", 2015.

[13] Ethan D. Bloch, "Polygons, Polyhedra, Patterns & Beyond", 2015.

تطبيق خوارزمية *A لتخطيط المسار على تحليل C-SPACE المعدل

فراس عبد الرزاق رحيم* أسماء عبد اللطيف حسين**

*،**قسم هندسة السيطرة والنظم/ الجامعة التكنولوجية

*البريد الالكتروني: 60124@uotechnology.edu.iq

**البريد الالكتروني: asmaa.alshawi@yahoo.com

الخلاصة

في هذا البحث، تم تقديم اشتقاق مستحدث لتحليل بنية (C-space). الفائدة من استخدام (C-space) هي لجعل عملية تخطيط المسار أكثر امان وسهولة. بعد الحصول على بنية وخارطة (C-space) لذراع الروبوت ثنائي الذراع، والتي تتضمن كل احتمالات الاصطدام بين أجزاء الروبوت والعوائق المحيطة به، خوارزمية *A، والتي عادة يتم تطبيقها في مجال العمل الكارتيزي للروبوت لإيجاد المسار المخمن، تم تطبيقها لإيجاد المسار المخمن على خارطة (C-space). بضع تعديلات لازمة لتطبيق طريقتنا المطورة على ذراع روبوت ذي حريات حركة أكثر من اثنين. نتائج خارطة (C-space)، والتي تم اشتقاقها بطريقتنا المطورة، تثبت دقة الخارطة الكلية (C-space) وبنيته، ومن ثم مسار ناجح ومضمون من نقطة البداية الى نقطة الهدف تم الحصول عليه بدون أي احتمالية اصطدام. تم الحصول على النتائج بوساطة برنامج (Matlab R2015a) باستخدام حاسبة بالمواصفات التالية: Intel (R) Core (TM) i3-3120M CPU.