Robotica: (2020) volume 38, pp. 1176–1190. © Cambridge University Press 2019
doi:10.1017/S0263574719001310

A Novel Collision-Free Path Planning Modeling and Simulation Methodology for Robotical Arms Using Resistive Grids

Carlos Hernández-Mejía†*, Héctor Vázquez-Leal‡ and Delia Torres-Muñoz‡

†*Escuela de Ingenierías, Universidad de Xalapa, Xalapa, Mexico*
‡*Maestría en Ingeniería Electrónica y Computación, Universidad Veracruzana, Xalapa, Mexico*
E-mails: hvazquez@uv.mx, deletsm@gmail.com

(Accepted July 31, 2019. First published online: August 22, 2019)

SUMMARY
Path planning represents planning collision-free strategies to move from starting point to ending point. These strategies can be carried out for known and unknown environments. Recently, a novel and reduced CPU-time modeling and simulation methodology for path planning in known environment based on resistive grids (RGs) has been introduced. In this work, a novel modified version of Resistive Grid Path Planning Methodology (RGPPM) methodology is presented with the purpose of exploring collision-free path planning for robotic arms. This extension of the methodology allows to numerically relate positions in the RG with angular values of the robotic systems. In addition, it is possible to include obstacles in the configuration space, and therefore collision detection can be established for RGs. Finally, the variation of links for robotic arms and obstacles for configuration space is explored by simulating different scenarios.

KEYWORDS: Resistive grids; Path planning methods; Robotical arms; Grid-based methods.

1. Introduction

Robotics[1,2] has been considered as a multidisciplinary area that with other emerging technologies such as information technology, biotechnology, and nanotechnology have contributed to increase opportunities for economic growth and significantly affect future generations with substantial social and economic impacts. Currently, robotics impacts in numerous fields as the amount of tasks carried out by robotical systems increases progressively, such as industrial, medical, services, military, and entertainment.

Industrial robots have been taken into account as a cornerstone of competitive manufacturing because they aim to combine high standards of productivity, quality, and adaptability with minimum costs. Nowadays, there are companies that provide robotic systems for industries, such as telecommunications, electronics, automotive, and pharmaceutical products. These industrial technological developments, together with advances in research laboratories, have profoundly driven robotics in various sectors of the technical world.

Robotic manipulators and robotic arms have been the first robotic systems in the industry, and therefore they have been placed in fundamental positions due to the fact that much of the technology to make robots more human-friendly and adaptable for various applications have emerged from manufacturing companies of industrial robots. In addition, based on the research needs in design,

* Corresponding author. E-mail: cmahernandez@gmail.com

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

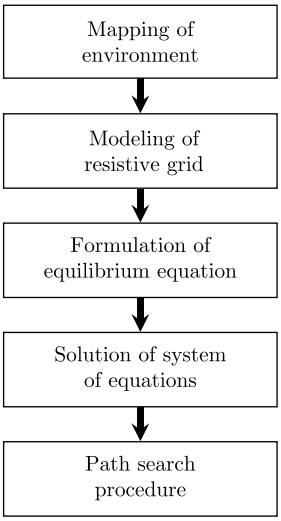


Fig. 1. Modeling and simulation methodology.

modeling, motion planning, and control for actual physical realizations of a robot, such as arms, legs, and hands, it is possible to understand that there are still a lot of problems to solve.

According to Japan Industrial Robot Association (JIRA), industrial robots can be classified as manual robots, fixed sequence robot, variable sequence robot, numerical robot, playback robot, and intelligent robot. In this work, a novel collision-free path planning methodology for robotical arms is introduced in order to overcome new challenges in fixed and variable sequence robots.

Robotic arms use mathematical algorithms for planning collision-free trajectories from starting point to ending point which is generally known as path planning. Path planning is the process of designing geometric (kinematic) specifications of the positions and orientations for robotic systems. It has been considered as a cornerstone in many research fields such as robotic navigation, computer graphics, surgical simulation, and computer-aided design applications.

During the last 30 years, path planning problem has been exhaustively tackled. This research area has produced considerable theoretical and practical solution proposals, such as grid-based, geometrical-based, and sampling-based methods. Grid-based approaches overlay a grid on configuration space, and they assume that there is a grid point for each location in the configuration space, for example, reconfigurable mesh (RMESH),[3] grid roadmap (GRM),[4] and resistive grid (RG).[5–12] Geometrical-based algorithms generate position and orientation trajectories for geometrically complex spatial paths using mathematical expressions, for example, ordinary differential equation-based algorithm,[13] parametric parabolas,[14] homotopy continuation method,[15,16] angle preprocessing,[17] and visibility binary tree algorithm.[18] Finally, sampling-based algorithms represent configuration space with roadmaps of sampled configurations, for example, rapidly exploring random tree (RRT) algorithm,[19–21] evolutionary optimization algorithm,[22] co-evolutionary improved genetic algorithm (CIGA),[23] neural computation,[24] and multi-objective particle swarm optimization.[25] In this work, grid-based method has been chosen in order to plan collision-free paths for robotical arms.

The manuscript is organized as follows. Section 2 introduces the novel collision-free path planning using RGs. Simulation results and discussion are presented in Sections 3 and 4, respectively. Finally, in Section 4, some conclusions are drawn.

2. Modeling and Simulation Methodology

In this part of the manuscript, a novel collision-free path planning modeling and simulation methodology for robotical arms is introduced. This methodology is shown in Fig. 1, and it is compounded of five stages: mapping of environment, modeling of RG, formulation of equilibrium equations, solution of system of equation, and path search procedure.

2.1. Mapping of environment

Mapping of the configuration space for the robotic arm is the first stage in the methodology. In order to carry out this task it is necessary to characterize the robotic arm and the configuration space. Figure 2 shows the fundamental elements of the configuration space and the robotic arm.

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

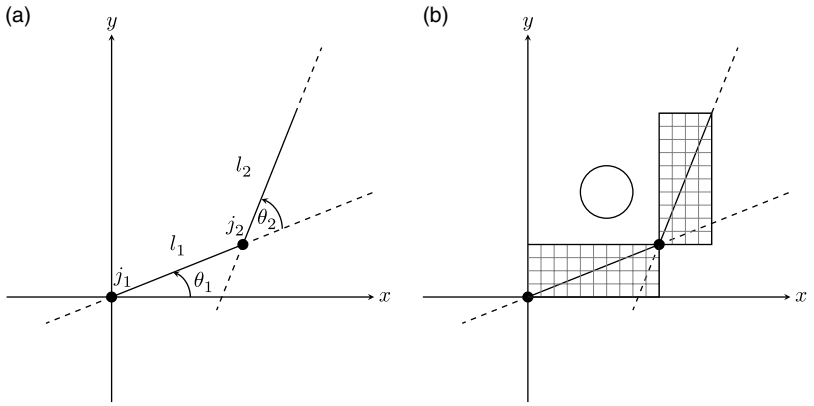


Fig. 2. Characterization for robotic arm with two links.

According to Fig. 2a, the robotic arm consists of two links, called l_1 and l_2. The links are represented by mathematical expressions as

$$y - y_0 = (x - x_0)\tan\theta \tag{1}$$

where x_0 and y_0 are xy-coordinates of the join j in the link and θ represents the angle of the link with respect to the horizontal line. It is important to mention that the angle θ for the first link of the robotic arm has been measured with respect to the horizontal line of the xy-plane. Moreover, the angle θ for the second link of the robotic arm has been measured with respect to the projection of the straight line of the previous link. Figure 2a shows the characteristics of the angles (θ_1 and θ_2), joints (j_1 and j_2), and links (l_1 and l_2) of the robotic arm.

In addition, Fig. 2b shows the local configuration spaces for the links, drawn by grid zones, and the existence of obstacles, represented by circles. Mathematical expressions for obstacles are established as

$$(x - h)^2 + (y - k)^2 = r^2 \tag{2}$$

where h and k are xy-coordinates of the center of the circle and r is the radius of the circle. It is important to note that the local configuration spaces exist because the links have finite lengths which collides or does not collide with the obstacles in a range of xy-coordinates. Besides, the local configuration spaces for the links change as the angles increase or decrease.

Given that fundamental elements of robotic arms and configuration spaces have been established, combinations of prohibited and non-prohibited angles must be determined, that is, angular combinations where the links of the robotic arm collide or do not collide with the obstacles located in the configuration space. In order to do this, it is necessary to rotate the links 360° using step-sizes according to a minimum angular resolution with the purpose of knowing if the link intercepts obstacles in some angular value.

Classical procedure to know if the link intercepts obstacles is based on evaluating the existence of real solutions, coming from the system equation constituted by (1) and (2), in the local configuration space as the link changes its angular value. In the search for colliding angular values, three different cases have been explored in Fig. 3.

In the first case, the system of equations composed of link and obstacle does not contain real solutions, that is, there is no collision, as shown in Fig. 3a. It is important to mention that nonexistence of real solutions covers not only the local configuration space of the link but also the projections on the straight line of the link.

For the second case, the solution coming from the system of equations represented by link and mathematical expression of obstacle is real but it is not found within the local configuration space of the link, as shown in Fig. 3b. This means that the existence of real solutions arises from interceptions with the projection of the link.

Finally, in the third case, the mathematical equations of link and obstacle constitute a system of equations with real solutions which belong to the local configuration space of the link, as shown in Fig. 3c.

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

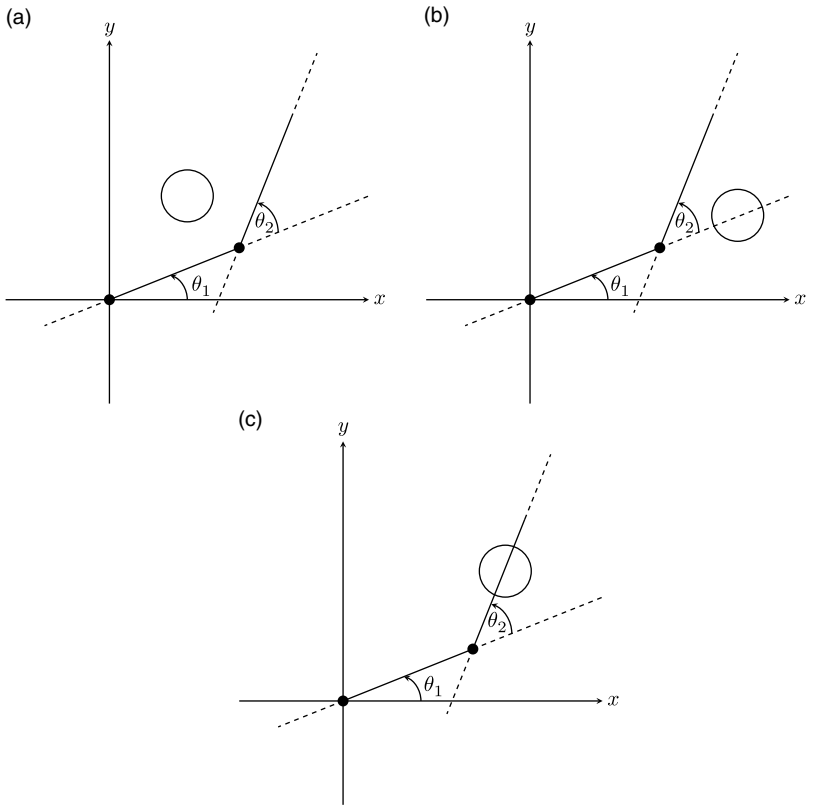


Fig. 3. Three different cases for searching of prohibited and non-prohibited angular values.

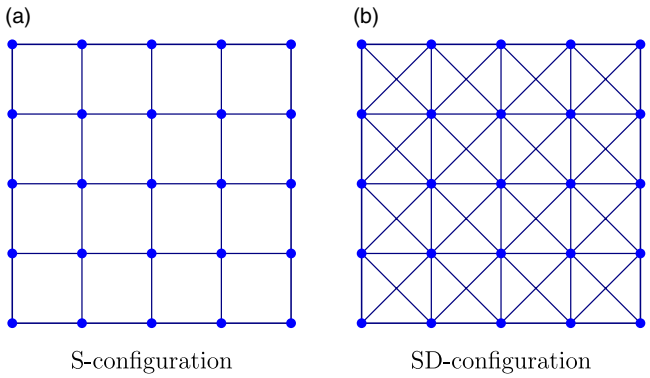


Fig. 4. Geometrical grid configurations.

According to the previous cases, the first and second cases cause angular values free of collisions with obstacles. On the other hand, in the third case, angular positions of the link collide with obstacles.

In the moment that prohibited and non-prohibited angular values have been established, it is necessary to transfer this data to RGs.

2.2. Modeling of RG

In this second stage, the modeling of the configuration space using RGs is carried out. It can be implemented using different geometrical arrangements, as shown in Fig. 4. These configurations are: square (S) and square with diagonals (SD).

According to the geometrical configurations of Fig. 4, the distribution of the resistors can be implemented with different orientations and values. RG configurations are shown in Fig. 5.

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

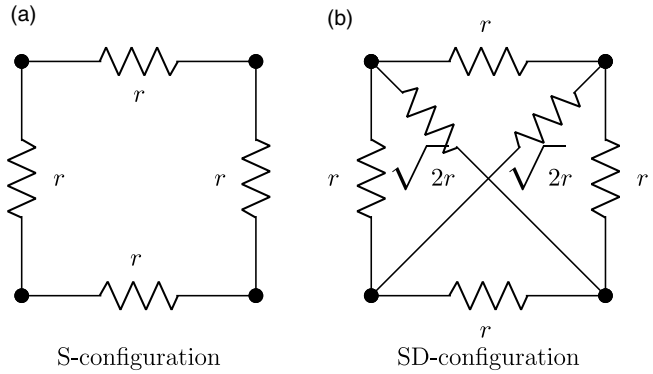


Fig. 5. Resistive grid configurations.

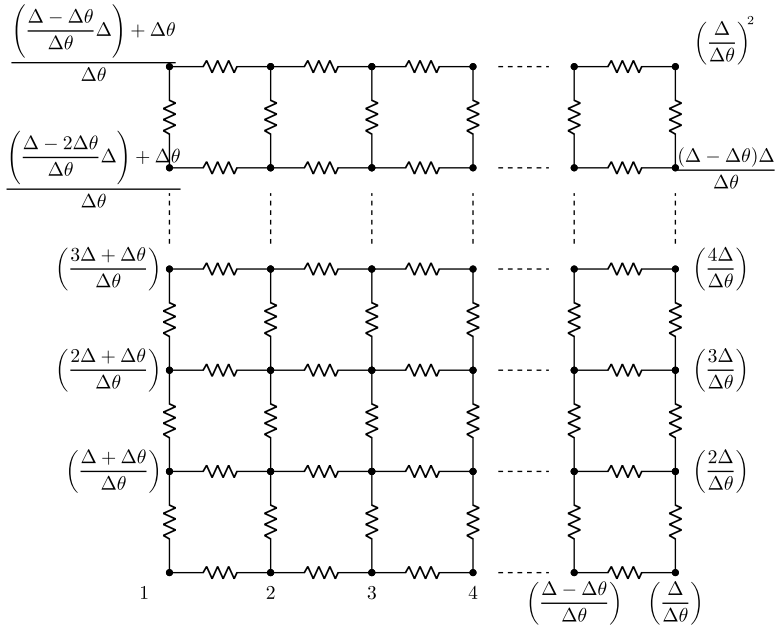


Fig. 6. Nodal distribution in resistive grid.

It is important to notice that the values of the resistors are related to the length of the geometrical configuration. In this order of ideas, by using fundamental geometry, it has been established that S-configuration contains resistors with homogeneous values. However, this is not possible for the SD-configuration because the values of the resistors located at the edge of the resistive configuration are not homogeneous with the resistors placed within the resistive configuration. This situation is shown in Fig. 5.

At the moment that geometrical arrangement of the configuration space has been established, it is necessary to transfer angular combinations of prohibited and non-prohibited angles towards nodes of the RG, that is, angular combinations must be related to numerical node in the RG. In order to do this, angle-node relation is governed by the following mathematical expression:

$$n = \left(\left(\frac{\theta_1}{\Delta\theta} - 1\right)\frac{\Delta}{\Delta\theta}\right) + \frac{\theta_2}{\Delta\theta} \tag{3}$$

where θ_1 and θ_2 are first and second link angles, respectively. $\Delta\theta$ is the minimum angular resolution, Δ is the maximum angle of rotation of the links, and n is the numerical node of the angular combination (θ_1, θ_2). In this section of the methodology, numerical values of $\Delta\theta$ can be [1° 2° 5° 10°]. In addition, Δ has been set numerically to 360° for all the links of the robotical arm. Figure 6 shows the nodal distribution of the RG according to the numerical values of Δ and $\Delta\theta$.

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

In the case of incorporating obstacles in the configuration space, the RG configuration must be modified by eliminating the nodes that belong to the obstacles. This removal of connections generates gap areas in the RG, that is, free-nodes zones.

According to (3), angular combinations of prohibited angles are transformed into prohibited nodes in the RG. It is important to emphasize that removing prohibited nodes causes the elimination of all resistors connected to them.

When the dimensions of the configuration space and the RG have been established, it is necessary to define the start point and the end point of the trajectory. At the start point, the power supply (voltage or current mode) must be connected and the ground must be connected at the end point.

2.3. Formulation of equilibrium equation

With a properly modeled environment, it is possible to formulate the equilibrium equation of the circuit using the Kirchhoff's laws. In order to do this, the modified nodal analysis (MNA)[26] has been resorted because it is an industrial standard for circuit simulation programs.[27,28]

The MNA procedure consists of formulating the equilibrium equations of the RG configurations by using the element's branch constitutive equations. These MNA descriptions are called stamps. In the particular cases of the RG configurations shown in Fig. 5, the conductance and independent voltage source (IVS) stamps have been required.

The conductance element yields the stamp

$$\begin{array}{c|cc|c} & v_i & v_j & \text{RHS} \\ \hline i & G & -G & 0 \\ j & -G & G & 0 \end{array} \tag{4}$$

where v_i and v_j represent the nodal voltage with respect to ground, i and j are the nodes of the element, $G = \frac{1}{R}$ is the equivalent conductance, and RHS means the right-hand side of the equilibrium equation. As it can be seen, the equivalent conductance has been resorted because its stamp is simpler than the resistance's stamp, and it does not add extra variables to the equilibrium equation.

The IVS yields the stamp

$$\begin{array}{c|ccc|c} & v_i & v_j & i_E & \text{RHS} \\ \hline i & 0 & 0 & +1 & 0 \\ j & 0 & 0 & -1 & 0 \\ \text{BE} & +1 & -1 & 0 & E \end{array} \tag{5}$$

where v_i and v_j represent the nodal voltage with respect to ground, i and j are the terminal nodes of the IVS, E is the supplied voltage by the IVS, and i_E is the unknown current that crosses by the IVS. It is important to note that the IVS is not compatible with the nodal analysis, and therefore it is necessary to add an extra column and row to the matrix representation. This extra column corresponds to i_E and the extra row corresponds to the branch equation (BE) of the IVS.

According to Figs. 4, 5, and 6, the configuration spaces can be considered as RGs of $(\Delta/\Delta\theta)^2$ size. In this manuscript, the shape of the configuration space has been considered as squares with side-length $(\Delta/\Delta\theta)$, where $(\Delta/\Delta\theta)$ means the amount of nodes on each side of the RG.

2.4. Solution of system of equations

As it can be seen in the following mathematical expression:

$$\Delta n = \left(\frac{\Delta}{\Delta\theta}\right)^2 \tag{6}$$

where Δn is the amount of total nodes in the RG. It is important to notice that the assembling of high density RGs potentially causes large sparse linear system of equations. Therefore, the solution of the configuration spaces needs to be carried out using a direct method for solving systems of linear equations in the form of $Ax = b$, where A is a sparse matrix (MNA matrix) and x and b are vectors or matrices (unknowns and stimulus vector, respectively).

In this manuscript, Multifrontal method from UMFPACK[29] has been resorted for solving the system of equations because it organizes the operations that take place during the factorization of sparse

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

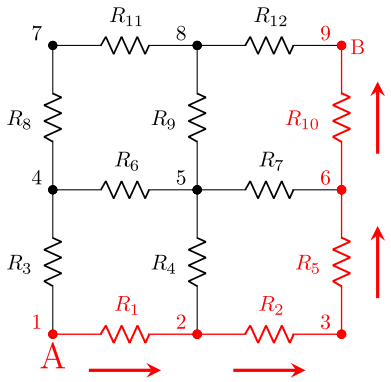


Fig. 7. General procedure for LCC algorithm.

matrices in such a way that the entire factorization is performed through partial factorizations of a sequence of dense and small submatrices causing a notable reduction in CPU-time. Moreover, it is capable of solving the formulation of equilibrium equations in order to compute the nodal and the branch electrical variables of the RG.

2.5. *Path search procedure*

Path search procedure consists of finding the path from the power supply to the ground (target) using the numerical results from the formulation and solution of equilibrium equations of the RGs. In order to do this, the Local Current Comparation (LCC) algorithm[30] has been implemented as shown in Fig. 7.

This algorithm starts the search of the path at the starting node (A). First, it determines the branches connected to the actual node. Then, it calculates the branch currents using the following expression:

$$I_j = \frac{V - V_j}{R_j}$$

where I_j, V, V_j, and R_j represent the current in the j-branch, the actual nodal voltage, the nodal voltage of the next node in the j-branch, and the resistance value in the j-branch, respectively.

When branch currents have been calculated, the LCC algorithm determines the highest branch current. After that, the branch with the largest current value is used as trajectory to the new actual node in order to continue the searching process until the ending node (B) is reached. Finally, the nodes under evaluation are considered points of the principal path, and therefore the resistors with the largest currents are considered branches of the principal path.

It is important to mention that the LCC algorithm uses the largest branch current in the node under evaluation in order to cross the branch to the next actual node, and therefore it is impossible to traverse the same branch more than once.

At the moment that the ending node is reached through the path search procedure, this part of the methodology determines the nodes that constitute the complete trajectory, and therefore it is possible to calculate the angular combinations of the links by using the following mathematical expressions:

$$\theta_1 = \begin{cases} (Z_1 + 1)\Delta\theta & \text{if } \dfrac{n\Delta\theta}{\Delta} \geq 1 \\ \\ (Z_2 + 1)\Delta\theta & \text{if } \dfrac{n\Delta\theta}{\Delta} < 1 \end{cases} \tag{7}$$

where Z_1 and Z_2 represent the numerical values of integer parts in $\frac{n\Delta\theta}{\Delta+\Delta\theta}$ and $\frac{n\Delta\theta}{\Delta}$, respectively.

$$\theta_2 = n\Delta\theta - \Delta\left(\frac{\theta_1}{\Delta\theta} - 1\right) \tag{8}$$

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

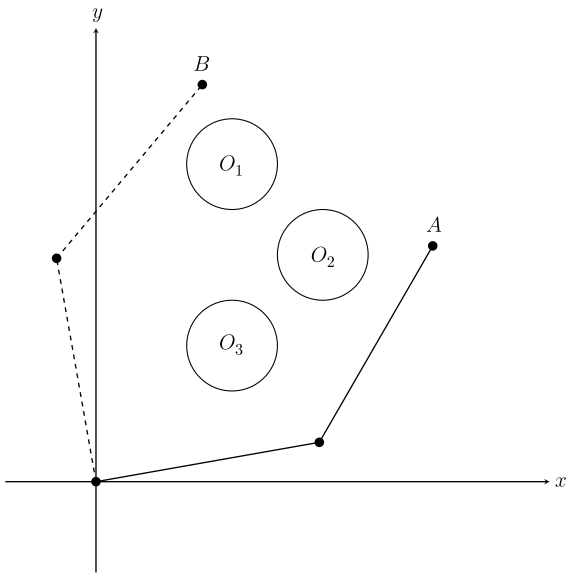


Fig. 8. 2-DOF Robot arm configuration space in first scenario.

3. Simulation Results

In this part of the manuscript, the novel methodology has been explored using three different scenarios. These scenarios have been established with the purpose of evaluating the behavior of the robotic arms as the amount of links and obstacles increases.

3.1. 2-DOF robotic arm with three obstacles

In the case of the first scenario, the starting and ending angular combinations have been established in the *xy*-coordinates plane as shown in Fig. 8.

As it can be seen in Fig. 8, the starting angular combination A of the robotic arm must move to ending angular combination B by varying θ_1 and θ_2 angles of l_1 and l_2 links, respectively. In order to do this, it is necessary to find a collision-free path because there are three obstacles O_1, O_2, and O_3 in the configuration space of the robotic arm.

According to (1) and (2), the links are characterized by the following mathematical expressions:

$$\begin{gathered} y = x \cdot \tan\theta_1 \\ y - (L \cdot \sin\theta_1) = (x - (L \cdot \cos\theta_1)) \tan(\theta_1 + \theta_2) \end{gathered} \tag{9}$$

where L means the length of the second link.

In addition, the obstacles are represented by

$$\begin{gathered} (x-3)^2 + (y-7)^2 - 1 = 0 \\ (x-5)^2 + (y-5)^2 - 1 = 0 \\ (x-3)^2 + (y-3)^2 - 1 = 0 \end{gathered} \tag{10}$$

and the numerical values of A and B positions are $A(10°, 50°)$ and $B(100°, 310°)$, respectively.

As it can be seen in Fig. 9, the configuration space has been established in the two-dimensional plane θ_1 and θ_2 where the bold dots represent prohibited angles, and therefore non-prohibited angles have been represented by collision-free areas in the angular combinations grid. Moreover, starting and ending positions have been located by using filled squares with red. Finally, collision-free path between A and B positions has been depicted by empty circles with blue.

It is important to mention that the path from A to B comes from carrying out the methodology in Fig. 1, and therefore the angular combinations of the path represent the nodes of the RG that connect to the branches with the greater amount of current. In addition, it is necessary to comment that the top and down limits in the θ_2 axis are interconnected. In the same way, the θ_1 axis interconnects

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

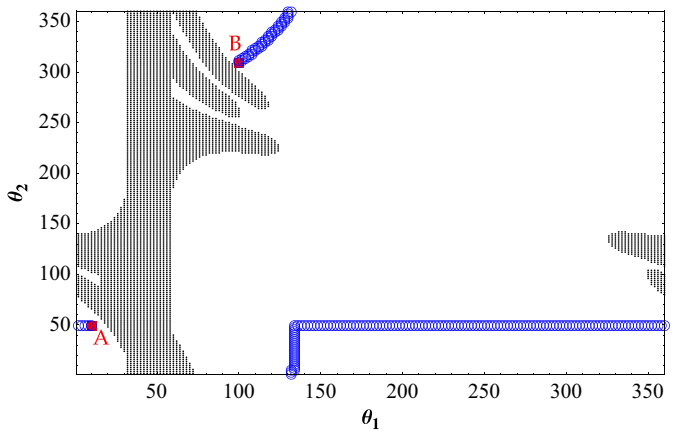


Fig. 9. θ_1 and θ_2 angular-combinations path in first scenario.

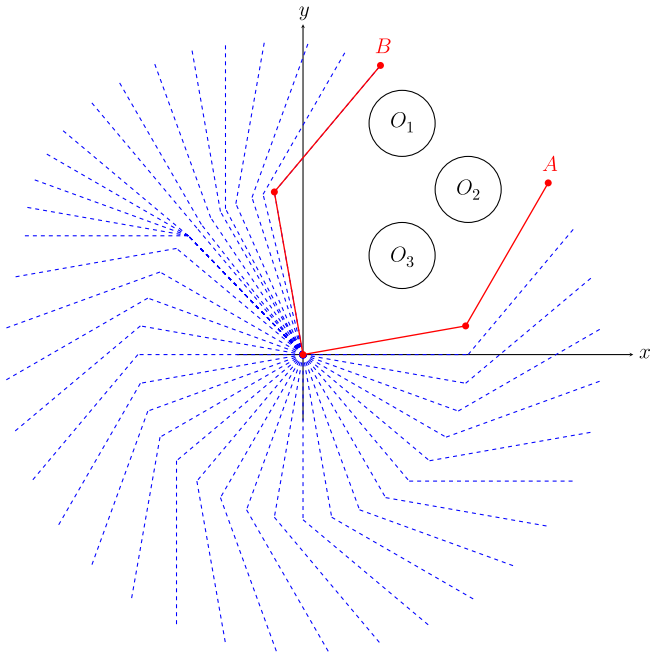


Fig. 10. 2-DOF Robot arm path in first scenario.

the limits of the left and right side. This explains how the path disappears from the left side and it emerges from the right side.

Figure 10 shows the path of the robotic arm in the plane-*xy* in order to move from position *A* to position *B*. As it can be seen, the robotic arm starts the path towards position *B* by moving the first link through different quadrants of the plane-*xy*. These movements reflect that the links of the robotic arm have no constraints. It is important to mention that Figs. 9 and 10 show three principal movements: variation of θ_1 as θ_2 remains fixed, variation of θ_2 as θ_1 remains fixed, and consecutive movements of θ_1 and θ_2.

3.2. 2-DOF robotic arm with ten obstacles

For the second scenario, the amount of obstacles has increased, and the starting and ending angular combinations have been established in the *xy*-coordinates plane as shown in Fig. 11.

As it can be seen in Fig. 11, it is necessary to find a collision-free path from starting angular combination *A* to ending angular combination *B* because there are ten obstacles O_1-O_{10} in the configuration space of the robotic arm.

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

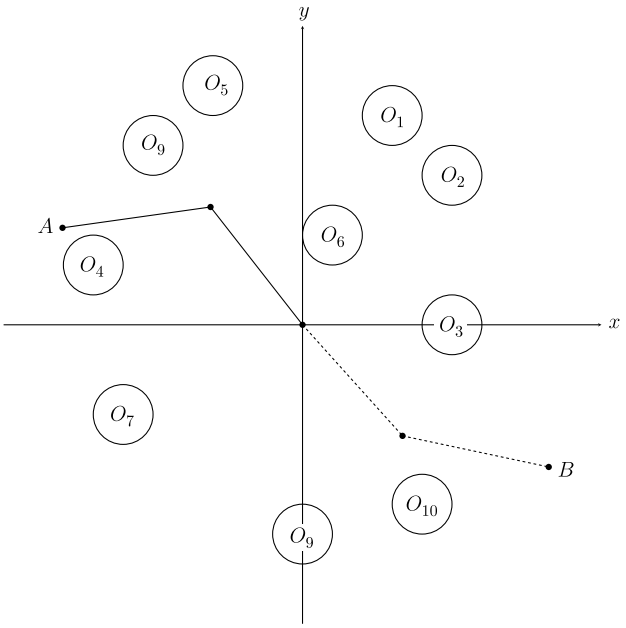


Fig. 11. 2-DOF Robot arm configuration space in second scenario.

In the same way as the first case, the mathematical expressions of links are represented by (9), and the obstacles are characterized by

$$\begin{aligned}
&(x-3)^2+(y-7)^2-1=0\\
&(x-5)^2+(y-5)^2-1=0\\
&(x-5)^2+(y-0)^2-1=0\\
&(x+7)^2+(y-2)^2-1=0\\
&(x+3)^2+(y-8)^2-1=0\\
&(x-1)^2+(y-3)^2-1=0\\
&(x+6)^2+(y+3)^2-1=0\\
&(x+0)^2+(y+7)^2-1=0\\
&(x+5)^2+(y-6)^2-1=0\\
&(x-4)^2+(y+6)^2-1=0
\end{aligned} \tag{11}$$

and the numerical values of A and B positions are $A(128°, 60°)$ and $B(312°, 36°)$, respectively.

As it can be seen in Fig. 12, collision-free path between A and B positions has been depicted by empty circles with blue. It is important to emphasize that the amount of obstacles and the positions of the prohibited angles reduce the configuration space available to generate the path from position A to position B. In addition, the path approaches to the shortest path because the prohibited angles are translated as prohibited nodes in the RG of the configuration space, and therefore the current distribution tends to focus on the smaller number of branches interconnected to the A and B positions.

Figure 13 shows the path of the links in the plane-xy in order to move from position A to position B.

As it can be seen in Fig. 13, the movements of the links when the robotic arm changes from position A to position B have been carried out near the obstacles because the configuration space for non-prohibited angles has been reduced.

3.3. 3-DOF robotic arm with three obstacles

In the third scenario, the amount of links has increased, and the starting and ending angular combinations have been established in the xy-coordinates plane as shown in Fig. 14.

As it can be seen in Fig. 14, there are three obstacles O_1–O_3 for the trajectory of the collision-free path from starting point A to ending point B. In addition, robotic arms have three links represented

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

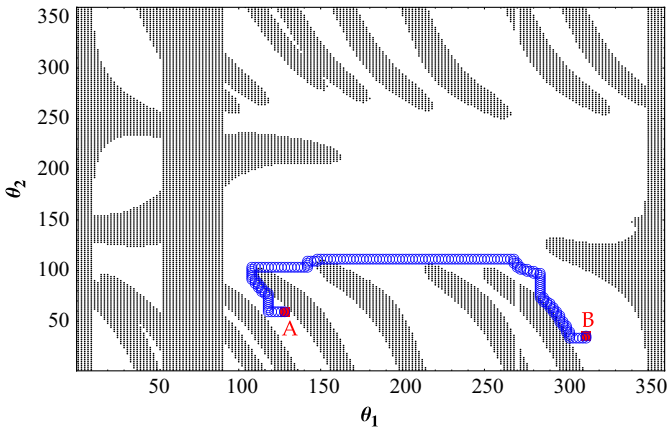


Fig. 12. θ_1 and θ_2 angular-combinations path in second scenario.

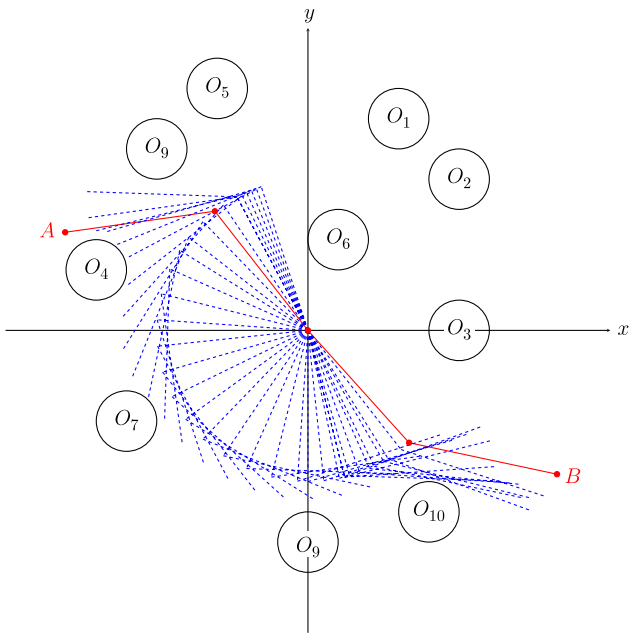


Fig. 13. 2-DOF Robot arm path in second scenario.

by the following mathematical expressions:

$$\begin{gathered} y = x \cdot \tan\theta_1 \\ y - (L_1 \cdot \sin\theta_1) = (x - (L_1 \cdot \cos\theta_1)) \tan(\theta_1 + \theta_2) \\ y - (L_2 \cdot \sin(\theta_1 + \theta_2)) = (x - (L_2 \cdot \cos(\theta_1 + \theta_2))) \tan(\theta_1 + \theta_2 + \theta_3) \end{gathered} \tag{12}$$

where L_1 and L_2 mean the length of the second and third links, respectively. Moreover, obstacles are represented by (10), and numerical values of A and B positions are $A(15°, 30°, 90°)$ and $B(90°, 300°, 300°)$, respectively.

As it can be seen in Fig. 15, the configuration space for the robotic arm with three links has been established in three dimensions θ_1, θ_2, and θ_3, where bold dots represent prohibited angles and therefore non-prohibited angles have been represented by collision-free areas in the angular combinations grid. Moreover, starting and ending positions have been located by using red dots. Finally, collision-free path has been depicted by blue dots.

In the same way as in the first case, the three-dimensional RG is interconnected between the starting and ending limits for each axis, and therefore it is possible that the path disappears and emerges on the sides of the cubical configuration space.

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

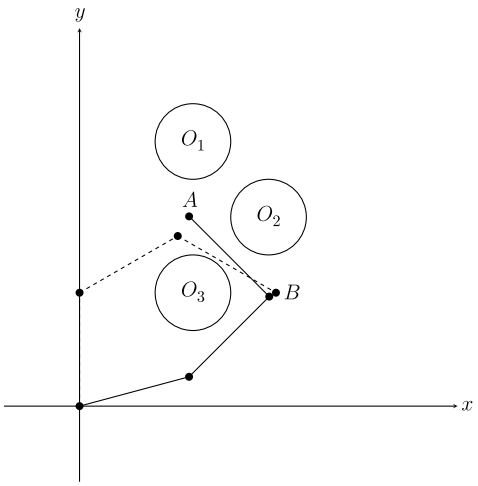


Fig. 14. 3-DOF Robot arm configuration space in third scenario.

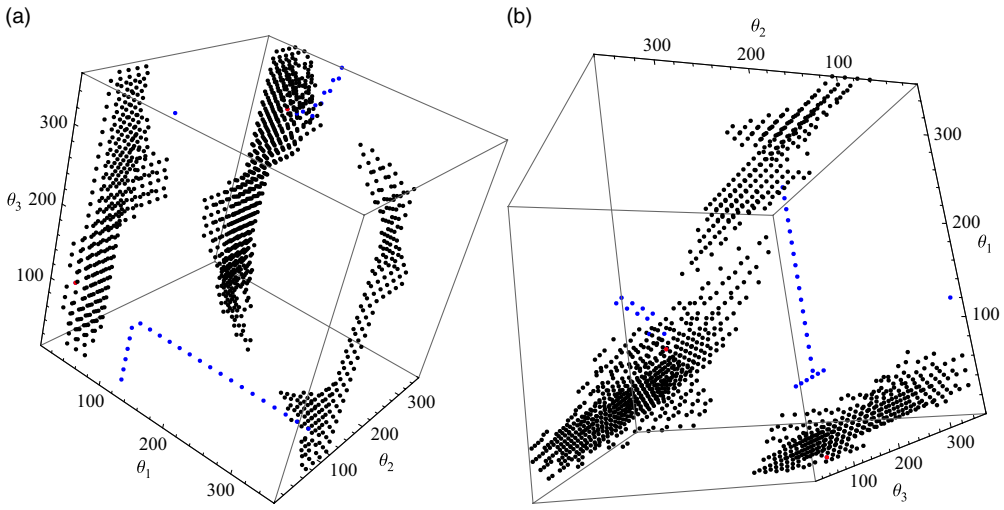


Fig. 15. θ_1, θ_2 and θ_3 angular-combinations path in third scenario.

Figure 16 shows the path of the links in the plane-*xy* in order to move from position *A* to position *B*. As it can be seen, the methodology is capable to find the path as the amount of links for robotic arm increases.

4. Discussion

According to mapping of environment stage, links have been considered as elements without mechanical restrictions for changing orientations. It means that as links are limited by angular ranges of operation, CPU-times decreases in order to establish prohibited and non-prohibited angles. Besides, as it has been proven in RGPPM methodology, this stage can be encoded using mid-level programming languages such as C and C^{++} or high-level programming languages such as Python in order to significantly reduce CPU-times.

As it can be seen in the simulation results section, RGs have been established for two and three dimensions. In the case of 2-DOFs robotic arm, the nodes of the two-dimensional RG are related to the coordinates (x, y) which represent angular combinations (θ_1, θ_2) of the links. For the case of 3-DOFs robotic arm, the angular combinations $(\theta_1, \theta_2, \theta_3)$ of the links have been related to the coordinates (x, y, z) which come from the nodes of the three-dimensional RG. Therefore, the RGPPM methodology is able to plan collision-free paths for robotic arms with higher degrees of freedom as long as *n*-dimensional resistive networks are built where *n* is the number of degrees of freedom of the robotic arm.

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

Table I. CPU-time for 2-DOF robotic arm with one obstacle.

Methodology stage	$\Delta\theta = 10$	$\Delta\theta = 5$	$\Delta\theta = 2$	$\Delta\theta = 1$
1	0.2500	0.8400	5.2200	19.3000
2	0.0032	0.0140	0.1000	0.4500
3	0.0046	0.0200	0.1400	0.6400
4	0.0110	0.0490	0.4500	3.8300
5	0.0010	0.0018	0.0048	0.0011

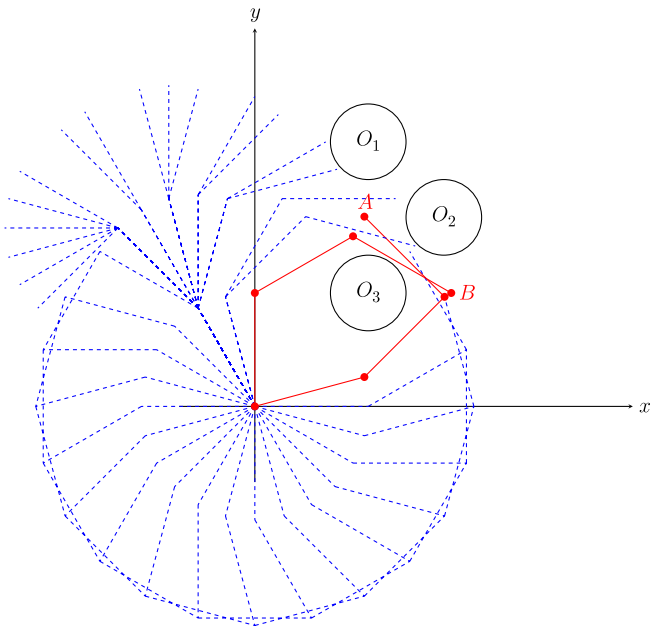


Fig. 16. 3-DOF Robot arm path in third scenario.

In ref. [31], probabilistic technique to incrementally build a roadmap in the configuration space of planar articulated robot is introduced. However, this technique does not guarantee complete connectivity in the path of the robotic system until it has had enough execution time. In addition, this method identifies hard regions, and it generates additional configurations to increase connectivity in the robotic system path. Moreover, it uses a pre-calculation process in order to recognize collisions with obstacles. Nevertheless, the principal drawback of probabilistic algorithms is producing a lot of iterative process over configuration space. This disadvantage grows exponentially as the size of the environment increases, and therefore computation time and memory usage increase considerably. In this work, collision-free path planning for robotic arms has been carried out using deterministic methodologies, and therefore the drawbacks of probabilistic methodologies do not exist.

In ref. [32], collision-free path planning for robotic arms is carried out using geometrical approaches for translating and rotating among obstacles. However, this method only uses I- and II-quadrant of Cartesian coordinate system in order to locate links and obstacles in the configuration space, and therefore it restricts the possibility of examining scenarios where obstacles are located in all quadrants of Cartesian plane. With the purpose of researching collision-free paths for robotic arms, this work opens the possibility of locating obstacles in the four quadrants of the configuration space.

In ref. [33], collision-free path planning for robotic arms using RRTs shows robust and effective simulation results. However, experimental setup lacks complex environment, that is, configuration spaces with large amounts of obstacles and narrow spaces. In this novel methodology, it ensures finding successful collision-free paths despite using environments with narrow spaces.

Based on the behavior of CPU-times shown in Table I, it can be observed that as the angular resolution decreases the CPU-time increases. This trend has been caused for increasing the number of mapping points in the environment, the number of nodes in the RG, and the dimensions of the MNA matrix. Moreover, it can be noticed that methodology stage mapping of environment presents

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

Table II. CPU-time for 2-DOF robotic arm with three obstacles.

Methodology stage	$\Delta\theta = 10$	$\Delta\theta = 5$	$\Delta\theta = 2$	$\Delta\theta = 1$
1	0.59000	2.2200	13.87000	54.760
2	0.00310	0.0120	0.09100	0.390
3	0.00390	0.0170	0.12290	0.550
4	0.00850	0.0350	0.32000	2.270
5	0.00014	0.0002	0.00053	0.001

considerable increases in CPU-time as the angular resolution decreases, and therefore it is the stage of the methodology that contributes with the highest execution time in order to find collision-free path for the robotic arm.

As it can be seen in Table II, the behavior of the CPU-time for each stage of the methodology is similar to the trend observed in Table I. However, if the execution times are compared in the second, third, fourth, and fifth stages of the methodology for the case of the configuration space with one obstacle against the execution times of the same stages for the case of the configuration space with three obstacles, it is possible to establish that CPU-time decreases as the number of obstacles increases in the configuration space of the robotic arm.

As far as the knowledge of the authors, there is no technique, method, or methodology in the literature that uses RGs in order to plan collision-free trajectories for robotic arms. In addition, the expansion of RGPPM for robotic arms consists of strategically adding the stage of environmental mapping in the principal structure of the methodology. This represents a significant advantage because it opens the possibility of transducing the angular information of the environment around the robotic arm to the construction of RGs to find collision-free paths using the electrical behavior of the RG. Therefore, we assure that the present work is a novel proposal for path planning in robotic arms.

Future research should be focused on the following topics:

(i) Coding the methodology in mid-level and high-level programming languages such as C, C^{++}, and Python. With the purpose of reducing computational resources: CPU-time and RAM memory usage.
(ii) Proposing innovative techniques to reduce computing times in the detection of obstacles.
(iii) Exploring the use of different devices into the configuration space, for example, memristive systems or Metal-Oxide Semiconductor (MOS) transistors.
(iv) Expanding the methodology to model and simulate robotic arms with greater degrees of freedom.
(v) Implementing the methodology for embedded system applications.

5. Conclusions

In this work, a novel collision-free path planning modeling and simulation methodology for robotical arms using RGs has been introduced. This research has been carried out using an expansion of RGPPM methodology capable of mapping the environment, modeling the configuration space, formulating and solving the equations of equilibrium, and searching for the trajectory from starting to ending point.

According to the simulation results, the methodology shows successful paths in order to move robotic arms from starting point to ending point without colliding with obstacles. Moreover, it has been demonstrated that trajectories continue to be successful despite of increasing the amount of obstacles in the configuration space or the amount of links in the robotic arm.

Finally, it can be concluded that the SC-configuration performs a convenient trade-off between the shortest path and the dimension of the linear system of equations.

References

1. B. Siciliano and O. Khatib, *Handbook of Robotics* (Springer, Berlin Heidelberg, 2008).
2. H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations* (MIT Press, Cambridge, MA, 2005).
3. D. Wang, "A linear-time algorithm for computing collision-free path on reconfigurable mesh," *Parall. Comput.* **34**(9), 487–496 (2008).

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press

4. M. R. Bahar, A. R. Ghiasi and H. B. Bahar, "Grid roadmap based ANN corridor search for collision free, path planning," *Scientia Iranica* **19**(6), 1850–1855 (2012).
5. L. Tarassenko and A. Blake, "Analogue Computation of Collision-Free Paths," *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA (1991) pp. 540–545.
6. M. Stan and W. Burleson, "Analog VLSI for Robot Path Planning," *Proceedings of the Twenty-Sixth Asilomar Conference on Signals, Systems Computers*, Pacific Grove, CA, USA (1992) pp. 915–919.
7. M. Stan, W. Burleson, C. Connolly and R. Grupen, "Analog VLSI for robot path planning," *Anal. Integr. Circ. Sig. Process.* **6**(1), 61–73 (1994).
8. K. Althofer, D. A. Fraser and G. Bugmann, "Rapid path planning for robotic manipulators using an emulated resistive grid," *Electr. Lett.* **31**(22), 1960–1961 (1995).
9. S. Koziol and P. Hasler, "Reconfigurable Analog VLSI circuits for robot path planning," *Proceedings of the NASA/ESA Conference on Adaptive Hardware and Systems*, San Diego, CA, USA (2011) pp. 36–43.
10. S. Koziol, P. Hasler and M. Stilman, "Robot Path Planning Using Field Programmable Analog Arrays," *Proceedings of the IEEE International Conference on Robotics and Automation*, Saint Paul, MN, USA (2012) pp. 1747–1752.
11. G. F. Marshall and L. Tarassenko, "Robot path planning using VLSI resistive grids," *Proceedings of the Third International Conference on Artificial Neural Networks*, Brighton, UK (1993) pp. 163–167.
12. G. F. Marshall and L. Tarassenko, "Robot Path Planning Using Resistive Grids," *Proceedings of the Second International Conference on Artificial Neural Networks*, Bournemouth, UK (1991) pp. 149–152.
13. M. S. Hassouna, A. E. Abdel-Hakim and A. A. Farag, "PDE-Based Robust Robotic Navigation," *Proceedings of the 2nd Canadian Conference on Computer and Robot Vision*, Victoria, BC, Canada (2005) pp. 176–183.
14. I. Namgung, "A global collision-free path planning using parametric parabola through Geometry Mapping of obstacles in robot work space," *KSME J.* **10**(4), 443 (1996).
15. H. Vazquez-Leal, A. Marin-Hernandez, Y. Khan, A. Yildirim, U. Filobello-Nino, R. Castaneda-Sheissa and V. M. Jimenez-Fernandez, "Exploring collision-free path planning by using homotopy continuation methods," *Appl. Math. Comput.* **219**(14), 7514–7532 (2013).
16. J. Karpinska and K. Tchon, "Continuation Method Approach to Trajectory Planning in Robotic Systems," *Proceedings of the 16th International Conference on Methods Models in Automation Robotics*, Miedzyzdroje, Poland (2011) pp. 51–56.
17. L. M. Ortega, A. J. Rueda and F. R. Feito, "A solution to the path planning problem using angle preprocessing," *Robot. Auton. Syst.* **58**(1), 27–36 (2010).
18. A. T. Rashid, A. A. Ali, M. Frasca and L. Fortuna, "Path planning with obstacle avoidance based on visibility binary tree algorithm," *Robot. Auto. Syst.* **61**(12), 1440–1449 (2013).
19. J. Nieto, E. Slawinski, V. Mut and B. Wagner, "Online Path Planning Based on Rapidly-Exploring Random Trees," *Proceedings of the IEEE International Conference on Industrial Technology*, Vina del Mar, Chile (2010) pp. 1451–1456.
20. A. Bry and N. Roy, "Rapidly-Exploring Random Belief Trees for Motion Planning Under Uncertainty," *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China (2011) pp. 723–730.
21. M. Kothari, I. Postlethwaite and D. W. Gu, "Multi-UAV Path Planning in Obstacle Rich Environments Using Rapidly-Exploring Random Trees," *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China (2009) pp. 3069–3074.
22. M. P. Aghababa, "3D path planning for underwater vehicles using five evolutionary optimization algorithms avoiding static and energetic obstacles," *Appl. Ocean Res.* **38**, 48–62 (2012).
23. Q. Hong, X. Ke and A. Takacs, "An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing* **120**, 509–517 (2013).
24. S. Lee and J. Park, "Neural computation for collision-free path planning," *J. Intell. Manufact.* **2**(5), 315–326 (1991).
25. Y. Zhang, D.-W. Gong and J.-H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing* **103**, 172–185 (2013).
26. C.-W. Ho, A. Ruehli and P. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circ. Syst.* **22**(6), 504–509 (1975).
27. A. F. Schwarz, *Computer-Aided Design of Microelectronic Circuits and Systems: General Introduction and Analog-Circuit Aspects* (Academic Press, Orlando, FL, USA, 1987).
28. J. Ogrodzki, *Circuit Simulation Methods and Algorithms* (Taylor & Francis, Abingdon, Oxfordshire, 1994).
29. T. A. Davis, "Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method," *ACM Trans. Math. Softw.* **30**(2), 196–199 (2004).
30. M. Kanaya, G. X. Cheng, K. Watanabe and M. Tanaka, "Shortest Path Searching for Robot Walking Using an Analog Resistive Network," *Proceedings of the IEEE International Symposium on Circuits and Systems*, London, UK (1994) pp. 311–314.
31. L. E. Kavraki, P. Svestka, J. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Auto.* **12**(4), 566–580 (1996).
32. R. T. Chien, L. Zhang and B. Zhang, "Planning collision-free paths for robotic arm among obstacles," *IEEE Trans. Pattern Anal. Mach. Intell.* **6**(1), 91–96 (1984).
33. M.-C. Kim and J.-B. Song, "Informed RRT* with improved converging rate by adopting wrapping procedure," *Intell. Serv. Robot.* **11**(1), 53–60 (2018).

https://doi.org/10.1017/S0263574719001310 Published online by Cambridge University Press