Trajectory Planning of a SCARA Manipulator using PRM for Collision Avoidance

Emerson V. A. Dias * **Catarina G. B. P. Silva** * **Josias G. Batista** *
Darielson A. Souza ** **José L. N. Silva** * **Geraldo L. B. Ramalho** *
Marcello A. F. B. Lima ***

* *Federal Institute of Ceará, Campus Fortaleza, CE, (e-mail: emersonverasifce@gmail.com, catarinagbezerra@gmail.com, josiasbatista@ifce.edu.br, leonardo.silva@ifce.edu.br, gramalho@ifce.edu.br).*
** *Department of Electrical Engineering, Federal University of Ceará, CE (e-mail: darielson@dee.ufc.br)*
*** *Federal Institute of Ceará, Campus Limoeiro do Norte, CE (e-mail: marcello@ifce.edu.br)*

Abstract: Robotics has grown a lot and more and more tasks are performed by robots. In many applications it is necessary that the robot does not stop if a collision is about to happen but that it deviates from this obstacle, be it a human being or another machine. This paper discusses the implementation of the Probabilistic Roadmap (PRM) algorithm in a SCARA manipulator (Selective Compliance Assembly Robot Arm). The algorithm is used in collision avoidance trajectory planning, for situations which will be compared by computational cost. The results show the trajectories generated by the algorithms in the Cartesian space and also the trajectories, speeds, accelerations and torques calculated from the dynamic model of each joint of the manipulator. The results of each situation are also presented, with circular and square obstacles and the number of points used in the simulation. In implementing the situation in which 100 points are used, the algorithm proved to be more efficient.

Keywords: Trajectory planning, probabilistic roadmap, collision avoidance path, SCARA manipulator, dynamic model.

1. INTRODUCTION

Robotics was introduced to the world in order to automate industrial procedures, that is, to facilitate human work. With the technological evolution, the use of robotics in the industrial environment has grown significantly (SOUZA, 2008). From this perspective, it is assumed that robots can perform a task with sufficient intelligence to classify what action is necessary to be chosen, for example, to avoid a collision.

The industrial sector uses devices to prevent accidents at work, for example, isolating an area so that the robot is able to work, and when an operator enters the robot's workspace, it can be served. However, it is more efficient to program the machine to understand the existence of an operator at the site. Therefore, the robot adopts a behavior in which it is safe, such as slowing down its movement or calculating a new trajectory. It is worth mentioning that the use of this practice eliminates the need to shut down the equipment, improving production efficiency and performance control (Wisskirchen et al., 2017).

The difficulty of finding an algorithm that allows the robot to move from $q_{initial}$ to q_{final} can be solved by the Probabilistic Roadmap (PRM) algorithm. This algorithm is a method based on a random sampling strategy that can solve the problem of effective paths that are difficult to construct with most algorithms in high-dimensional space (Cao et al., 2019). The PRM is proposed for the planning of paths and the identification of obstacles (Mohanta and Keshari, 2019), (Batista et al., 2020b)..

This work aims to implement the Probabilistic Roadmap algorithm applied to a SCARA (Selective Compliance Assembly Robot Arm) manipulator in the generation of collision avoidance trajectories with static obstacles. The implementation of the method is demonstrated from two maps, in which the algorithm is submitted to the challenge procedure of identifying the obstacles and elaborating a safe trajectory from the starting point ($q_{initial}$) to the final point (q_{final}). The result also shows the computational costs and the trajectories of the manipulator, as well as the speeds, accelerations and torques of the joints. Torques are calculated from the dynamic model that is also presented in this research.

2. SCARA MANIPULATOR

The SCARA manipulator is a 3-DOF robot, however, it will be used only 2 DOF. The robot is shown in Figure 1. Because it is a SCARA 3-DOF robot, the first two base joints are rotated around the vertical ax and thus it works in a horizontal plane (XY plane), therefore it operates as a planar 2-DOF robot.

Figure 1. Robotic manipulator SCARA.

2.1 Inverse Kinematics

Applying some trigonometric transformations we find the inverse kinematics equations, given by:

$$\theta_1 = tan^{-1}\left[\frac{P_y(L_1 + L_2 cos(\theta_2)) - P_x L_2 sen(\theta_2)}{P_x(L_1 + L_2 cos(\theta_2)) - P_y L_2 sen(\theta_2}\right] \quad (1)$$

$$\theta_2 = cos^{-1}\left(\frac{P_x^2 + P_y^2 - L_1^2 - L_2^2}{2L_1 L_2}\right) \quad (2)$$

where $L_1 = 0.35$ m and $L_2 = 0.30$ m, are the length values of each manipulator joint.

The equations (1) and (2) are the inverse kinematics problem solution for the SCARA manipulator. These equations are used to perform the manipulator's trajectory generation.

2.2 Dynamics of an industrial manipulator

In this, dynamic equations of movement for the manipulator will be derived. First, the kinetic and potential energy of the manipulator will be equated and then the Lagrange equation for the movement will be applied (Batista et al., 2020a), (Batista et al., 2018):

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau, \quad (3)$$

where $C(\theta, \dot{\theta}) \in \Re^n$ is the matrix that describes the centripetal and Coriolis forces, and $G(\theta) = \frac{\partial g}{\partial \theta} \in \Re^n$ is the gravity vector. The value adopted for gravity acceleration here was $g = 9.8\ m/s^2$ (Batista et al., 2018), (Batista et al., 2016).

2.3 Lagrange equation

Applying the Lagrange formulation to the system it will be possible to obtain the dynamics equations for joints 1 and 2 of the manipulator. Thereby We will obtain the first equation of motion that describes the torque of joint 1, which will be:

$$\begin{aligned}\tau_1 = [(m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 C_2]\ddot{\theta}_1 + \\ [m_2 l_2^2 + m_2 l_1 l_2 C_2]\ddot{\theta}_2 - 2m_2 l_1 l_2 S_2 \dot{\theta}_1 \dot{\theta}_2 - \\ m_2 l_1 l_2 S_2 \dot{\theta}_2^2 + m_2 g l_2 C_{12} + (m_1 + m_2) g l_1 C_1\end{aligned} \quad (4)$$

We can obtain the first equation of motion that describes the torque of joint 2, which will be:

$$\begin{aligned}\tau_2 = (m_2 l_2^2 + m_2 l_1 l_2 C_2)\ddot{\theta}_1 + m_2 l_2^2 \ddot{\theta}_2 + \\ m_2 l_1 l_2 S_2 \dot{\theta}_1^2 + m_2 g l_2 C_{12}\end{aligned} \quad (5)$$

Substituting the values of $l_1 = 0.35\ m$, $l_2 = 0.30\ m$, $m_1 = 7.872\ kg$, $m_2 = 4.277\ kg$ and $g = 9.8\ m/s^2$ in (4) and (5), we have:

$$\begin{aligned}\tau_1 = [1.873 + 0.898 C_2]\ddot{\theta}_1 + [0.384 + \\ 0.449 C_2]\ddot{\theta}_2 - 0.769 S_2 \dot{\theta}_1 \dot{\theta}_2 - \\ 0.384 S_2 \dot{\theta}_2^2 + 12.574 C_{12} + 41.671 C_1,\end{aligned} \quad (6)$$

and

$$\begin{aligned}\tau_2 = [0.384 + 0.449 C_2]\ddot{\theta}_1 + \\ 0.384\ddot{\theta}_2 + 0.449 S_2 \dot{\theta}_1^2 + 12.574 C_{12}.\end{aligned} \quad (7)$$

3. PROBABILISTIC ROADMAP METHOD

PRM randomly generates a set of configurations, which are represented as nodes. Then the planner connects these nodes until a more efficient path is elaborated (Spong et al., 2020). In the literature directed to PRM, these are the main functions used to determine the best connection between the created nodes, from the calculated distance:

$$||q' - q|| = [\sum_{i=1}^{n}(q_i' - q_i)^2]^{\frac{1}{2}}, \quad (8)$$

$$max_n |q_i' - q_i|, \quad (9)$$

$$[\sum p \epsilon A ||p(q') - p(q)||^2]^{\frac{1}{2}}, \quad (10)$$

$$max_p \epsilon A ||p(q') - p(q)||. \quad (11)$$

For the algorithm to identify the shortest path, distance calculations between nodes are performed. The (9) equation is the most used for this purpose (Spong et al., 2020).

3.1 Implementation of the PRM Algorithm

The logic of the Probabilistic Roadmap is based on previously analyzing the robot's trajectory from a predetermined map, therefore, this knowledge of the location that the robot will transit is possible for it to calculate the route it will follow. Therefore, the SCARA manipulator will calculate the route effectively, aiming to distance all obstacles along the path.

In the PRM, the map described for the machine will be analyzed, and verified, so that it is identified where obstacles and open access roads are. The method is based on random plotting of imaginary points. These points will be fixed in places that were determined to be free, that is, without obstacles. The trajectory developed by this algorithm values not only collision avoidance movement, but also the shortest path (Sciavicco and Siciliano, 2012), (Dias et al., 2021).

For trajectory planning with PRM, the following steps are necessary:

(1) The path is a graph $G(V, E)$;
(2) The robot configuration $q \rightarrow Q_{free}$ is a vertex;
(3) The edge $(q_1,\ q_2)$ implies a collision avoidance path between these robot configurations;
(4) A metric is required to $d(q_1,\ q_2)$ (for example, euclidean distance);
(5) Use of coarse knot sampling and fine edge;
(6) Result: a path in Q_{free}.

The pseudo-code of the Probabilistic Roadmap algorithm is presented below.

Algorithm Probabilistic Roadmap

Input:
n: number of input nodes in the roadmap
k: number of neighbors for each configuration
Output:
A roadmap $G = (V, E)$

1: $V \leftarrow \emptyset$
2: $E \leftarrow \emptyset$
3: **while** $|V| < n$ **do**
4: **repeat**
5: $q \leftarrow a$ random configuration in Q
6: **until when** q is collision free
7: $V \leftarrow V \cup \{q\}$
8: **end while**
9: **for all** $q \in V$ **do**
10: $N_q \leftarrow k$ neighbour's q chosen from V according to the distance
11: **for all** $q' \in N_q$ **do**
12: **if** $(q,\ q') \notin E$ **and** $\Delta(q,\ q') \neq$ null **then**
13: $E \leftarrow E \cup \{(q,\ q')\}$
14: **end if**
15: **end for**
16: **end for**

3.2 Scenario used

For the collision avoidance trajectory of the SCARA manipulator, the Cartesian space (XY) was considered. From the points obtained by the PRM algorithm, the inverse kinematics are solved, and the positions of each joint of the manipulator are found. These points can be applied to any manipulator that has a compatible workspace or to a mobile robot. The generated paths must avoid collision with two circular obstacles or with three square objects, shown in figures 2 and 3, with equal sizes whose radius is equal to 0.2 m and the side of the square is equal to 0.3 m.

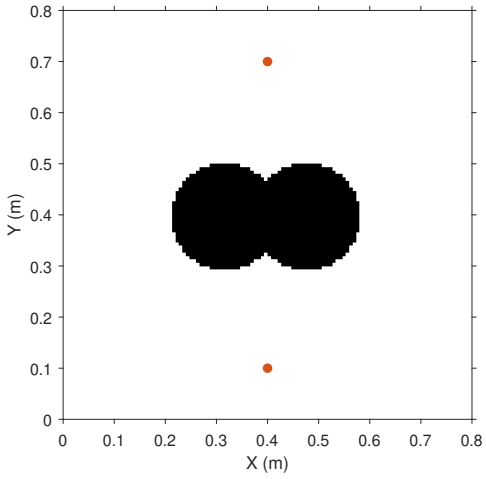

Figure 2. Scenario with circular obstacles for the implementation of the algorithm.

The scenario is formed by the starting position of (0.4; 0.7), ending position (0.4; 0.1) of the manipulator for circular obstacles and for square obstacles (0, 2768; 0.7810) for starting position and (0.672; 0.112) for final position, in Cartesian space. For the circular obstacles, positions (0.3; 0.4) were used for obstacle 1 (which

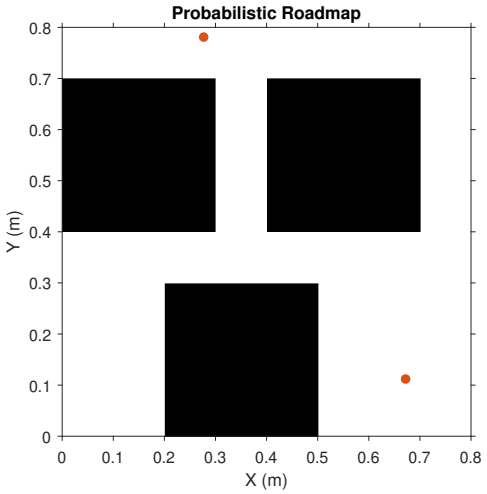

Figure 3. Scenario with square obstacles for the implementation of the algorithm.

is on the left most), and (0.5; 0.4) for obstacle 2 (which is on the far right). For the square obstacles the positions (0.19841; 0.55), (0.39671; 0.15) and (0.516255; 0.60) were used in the scene, positions that define their center in Cartesian space.

To evaluate the algorithm in each scenario, the processing time for each situation was used. The average values of these criteria are calculated for 20 repetitions of the simulation. The simulations are performed on a computer with a Core i3 processor - 7^{th} Generation, with a processing speed of 3.90 GHz and with RAM of 8.00 GB. The algorithm was implemented in the M-code language with some language specific functions.

4. RESULTS AND DISCUSSIONS

This section presents the results of the PRM algorithm, as well as the final considerations about the results. Situations are presented for the algorithm with 100 and 1000 points for square and circular obstacles. The trajectories, speeds, accelerations and torques of each joint of the manipulator under study are also presented.

4.1 PRM Results

Figures 4 and 5 shown the paths in Cartesian space with 100 and 1000 points for circular obstacles, respectively. It is observed that the PRM algorithm with 1000 points generates a collision path. Because the algorithm based on 1000 points has a small distance between the points and the generated points are closer to the obstacle. The PRM always tries to have the shortest path, so as the points were too close to the obstacles, a collision occurs.

Figures 6 and 7 shown the paths in Cartesian space with 100 and 1000 points using square obstacles. Note that the PRM with 1000 again generates a collision path.

4.2 Joints trajectories and torques

Figures 8, 9 and 10 shown the trajectories, speeds and accelerations, respectively, for circular obstacles, of joints

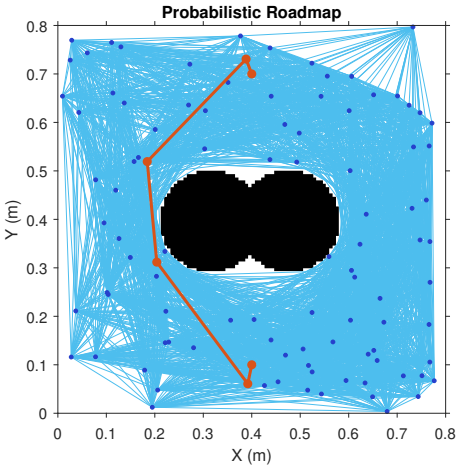

Figure 4. Path in the Cartesian space of the PRM algorithm with circular obstacles for 100 points.

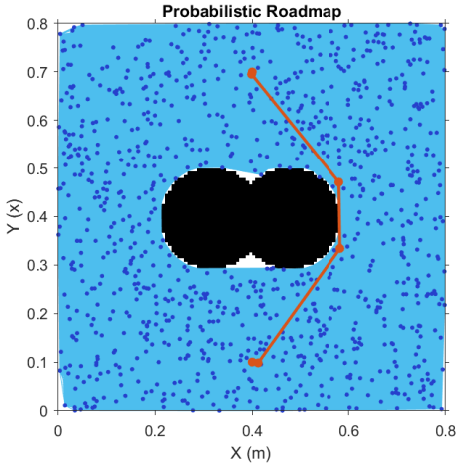

Figure 5. Path in the Cartesian space of the PRM algorithm with circular obstacles for 1000 points.

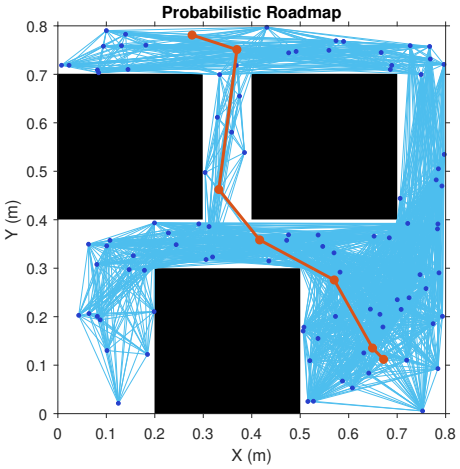

Figure 6. Path in the Cartesian space of the PRM algorithm with square obstacles for 100 points.

1 and 2, of the manipulator for the situation without collision, presented in Figure 4.

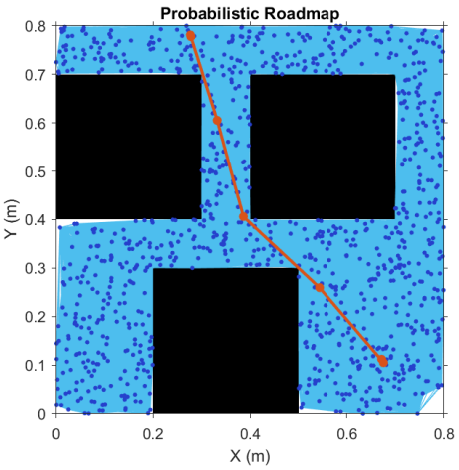

Figure 7. Path in the Cartesian space of the PRM algorithm with square obstacles for 1000 points.

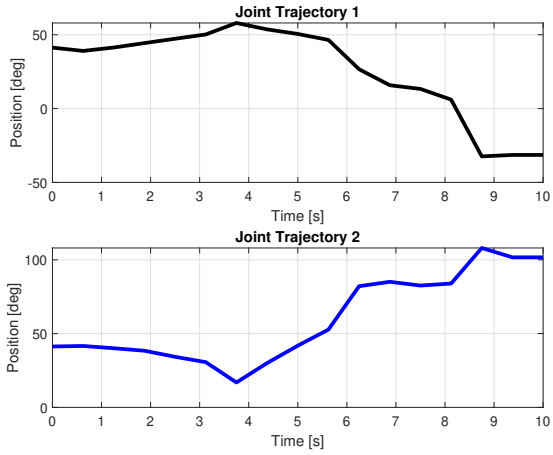

Figure 8. Trajectories of the SCARA manipulator joints generated from the PRM in the Cartesian space for circular obstacles.

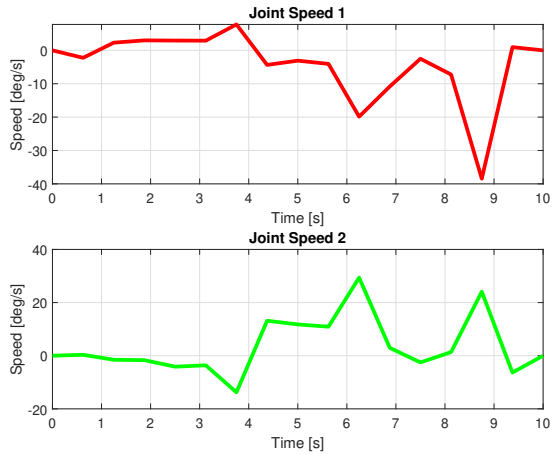

Figure 9. Joint space speeds for the PRM for circular obstacles.

Figure 11 shows the torques of the manipulator joints calculated from the dynamic model and the trajectories

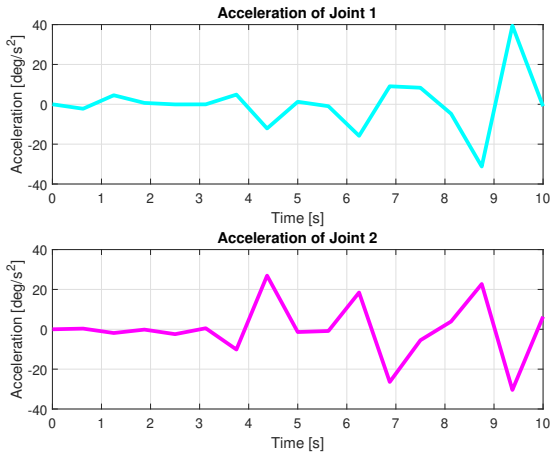

Figure 10. Accelerations in joint space for PRM for circular obstacles.

(Figure 8), speeds (Figure 9) and accelerations (Figure10) previously presented.

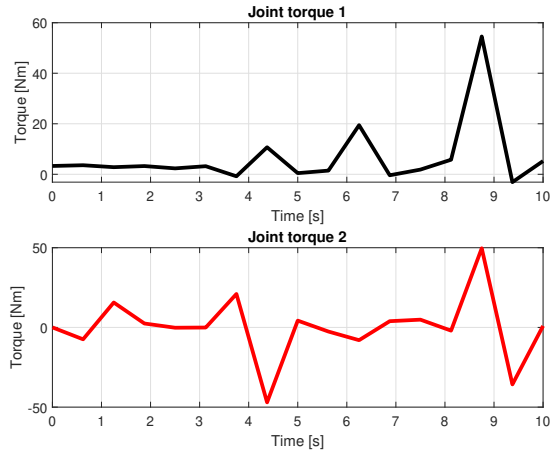

Figure 11. Torque of the joints for the trajectories, speeds and accelerations shown in figures 8, 9 and 10, respectively.

Figures 12, 13 and 14 shown the trajectories, speeds and accelerations, respectively, for square obstacles, of joints 1 and 2, of the manipulator for the situation without collision, presented in Figure 6.

Figure 15 shows the torques of the manipulator joints calculated from the dynamic model and the trajectories (Figure 12), speeds (Figure 13) and accelerations (Figure 14) previously presented.

4.3 Discussions

A more detailed comparison was made between the simulations of the algorithm. The average processing time for 20 repetitions of the simulation was verified for each situation that was worked on. Tables 1 and 2 show the comparisons of the processing of the algorithm for situations with circular obstacles and square obstacles, respectively.

The algorithm achieved better performance when applied to situations that use only 100 points to determine the

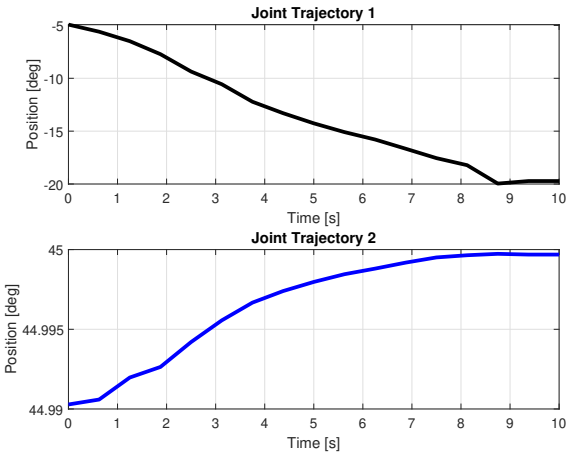

Figure 12. Trajectories of the SCARA manipulator joints generated from the PRM in the Cartesian space for square obstacles.

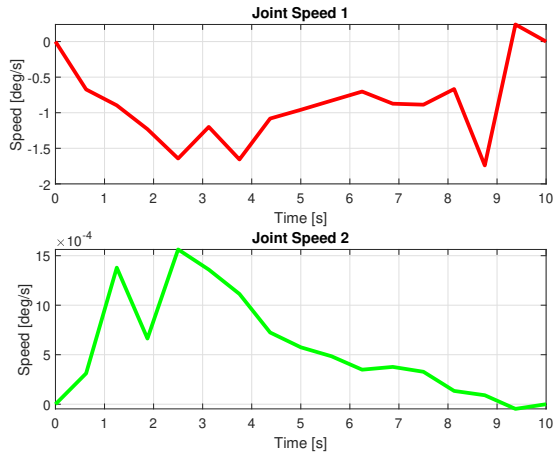

Figure 13. Joint space speeds for the PRM for square obstacles..

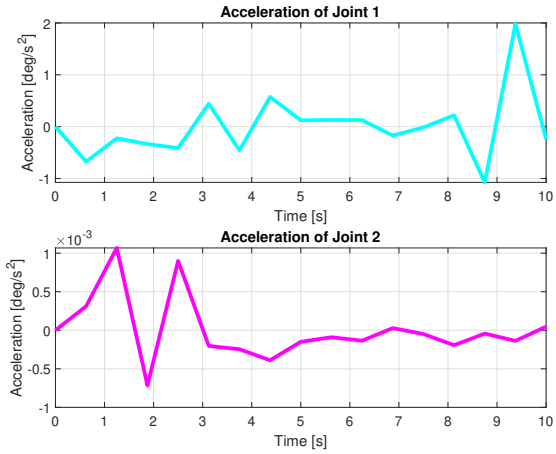

Figure 14. Accelerations in joint space for PRM for square obstacles.

trajectory. The trajectory with 100 points applied to the square obstacles was the one with the lowest average processing time. The importance of knowing the time

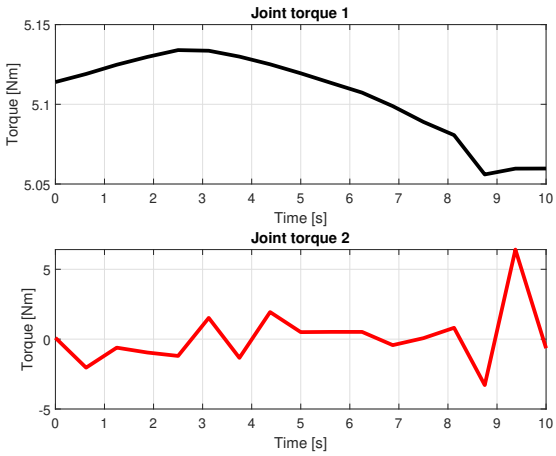

Figure 15. Torque of the joints for the trajectories, speeds and accelerations shown in figures 12, 13 and 14, respectively.

Table 1. Average processing time of the algorithm for circular obstacles.

No. of points	**Proc. Medium** $[s]$	**Trajectory** $[m]$
100	1.121537	0.88845
1000	1.641262	0.73370

Table 2. Average processing time of the algorithm for square obstacles.

No. of points	**Proc. Medium** $[s]$	**Trajectory** $[m]$
100	1.062079	0.85657
1000	7.871912	0.83825

spent to execute the algorithm is fundamental for an implementation online, that is, in real time, to verify if the algorithm is able to prevent obstacles in motion (dynamic obstacles). It is possible to analyze the simulation with 1000 points in both cases it has a shorter trajectory than in the simulations with 100 points. However, the simulation of 100 points with circular obstructions had better trajectory optimization and computational cost. For simulations with 1000 points, both had a collision, since with the addition of points it causes a trajectory very close to the obstacle.

5. CONCLUSIONS

The Probabilistic Roadmap algorithm is efficient in applications where collision avoidance trajectories are required, so it can be said that this study brings important contributions in this context. This paper used PRM to simulate collision avoidance navigation for certain types of situations. The PRM is based on random plotting of points, which served as a reference for determining the trajectory intended for the manipulator's movement. In the elaboration of the trajectory using the criteria used, all simulations that did not have collisions had low computational cost, for the situation with circular objects the time was 1.121537 s, while with square objects it was 1.062079 s.

In view of the results shown, it is concluded that the algorithm was satisfactory in preventing collision for the handler under study. The calculation of the torques of the joints are also important to know if it is possible for the manipulator to execute the trajectories with a certain torque value.

As future work we intend to: implement the algorithms to avoid collision in real time (online) and implement the Probabilistic Roadmap with computational intelligence.

AGRADECIMENTOS

The authors would like to thank the Institutional Program for Scientific Initiation Scholarships - PIBIC IFCE/ CNPq/ FUNCAP, Edital Nº 1/2020 - PRPI / Reitoria - IFCE, for the aid scholarship.

REFERENCES

Batista, J., Souza, D., dos Reis, L., Barbosa, A., and Araújo, R. (2020a). Dynamic model and inverse kinematic identification of a 3-dof manipulator using rlspso. *Sensors*, 20(2), 416.

Batista, J., Souza, D., Silva, J., Ramos, K., Costa, J., dos Reis, L., and Braga, A. (2020b). Trajectory planning using artificial potential fields with metaheuristics. *IEEE Latin America Transactions*, 18(05), 914–922.

Batista, J.G., da Silva, J.L., and Thé, G.A. (2018). Can artificial potentials suit for collision avoidance in factory floor?

Batista, J.G., da Silva, J.L., Pereira, N.S., et al. (2016). Modelagem matemática e simulação computacional da dinâmica de um robô scara. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 4(1).

Cao, K., Cheng, Q., Gao, S., Chen, Y., and Chen, C. (2019). Improved prm for path planning in narrow passages. In *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, 45–50. IEEE.

Dias, E.V., Silva, C.G., Batista, J.G., Ramalho, G.L., Costa, J.R., Silva, J.L., and Souza, D.A. (2021). Prevenção de colisão de um manipulador scara utilizando campos potenciais artificiais e caminhos probabilísticos. *Brazilian Journal of Development*, 7(1), 11252–11270.

Mohanta, J.C. and Keshari, A. (2019). A knowledge based fuzzy-probabilistic roadmap method for mobile robot navigation. *Applied Soft Computing*, 79, 391–409.

Sciavicco, L. and Siciliano, B. (2012). *Modelling and control of robot manipulators*. Springer Science & Business Media.

SOUZA, S.d. (2008). *Planejamento de trajetória para um robô móvel com duas rodas utilizando um algoritmo A-estrela modificado. 110 p.* Ph.D. thesis, Dissertação (Mestrado)—Universidade Federal do Rio de Janeiro.

Spong, M.W., Hutchinson, S., and Vidyasagar, M. (2020). *Robot modeling and control*. John Wiley & Sons.

Wisskirchen, G., Biacabe, B.T., Bormann, U., Muntz, A., Niehaus, G., Soler, G.J., and von Brauchitsch, B. (2017). Artificial intelligence and robotics and their impact on the workplace. *IBA Global Employment Institute*, 11(5), 49–67.