Path Planning Collision Avoidance of a SCARA Manipulator using PRM with Fuzzy ⋆

Emerson V. A. Dias * **Josias G. Batista** * **Catarina G. B. P. Silva** *
Geraldo L. B. Ramalho * **Darielson A. Souza** **
José Leonardo N. Silva * **André P. Moreira** ??

* *Mobile Robotics Laboratory - Department of Industry, Federal Institute of Education, Science and Technology of Ceará - IFCE, Campus Fortaleza, Fortaleza, CE, Brazil.(e-mail: emersonverasifce@gmail.com, josiasbatista@ifce.edu.br, catarinagbezerra@gmail.com, gramalho@ifce.edu.br, leonardo.silva@ifce.edu.br, apmoreira@ifce.edu.br).*
** *Research Group on Automation, Control and Robotics - Department of Electrical Engineering, Federal University of Ceará, Fortaleza, CE, Brazil (e-mail: darielson@dee.ufc.br)*

Abstract: Robotics was introduced in the world with the objective of automating industrial processes, that is, facilitating human work, and with this new algorithms have been emerging to increasingly improve the use of robots in an autonomous way. This paper compares the Probabilistic Road-map (PRM) and PRM-Fuzzy applied for a SCARA manipulator (Selective Compliance Assembly Robot Arm). The PRM-Fuzzy algorithm is used to optimize the path generated by the PRM, as it makes the path smoother and shorter. Are used in collision avoidance path planning and is compared by computational cost (processing time), multiple correlation coefficient, (R^2), the mechanical energy consumption of the motors, and the shortest collision avoidance paths. The results show the trajectories generated by the algorithms in the Cartesian space and also the trajectories of each joint of the manipulator, calculated from the inverse kinematics. Velocities, accelerations, and torques for trajectories are also shown. Several scenarios with different obstacles were used and in all cases, the PRM-Fuzzy algorithm performed better than the conventional PRM for the comparisons performed.

Keywords: Path planning, collision avoidance, probabilistic roadmap, SCARA manipulator, PRM-Fuzzy.

1. INTRODUCTION

Robotics was introduced in the world with the aim of automating industrial processes, that is, facilitating human work. With the technological evolution, the use of robotics in the applied industrial environment and, therefore, several types of robots were created with the purpose of assisting or even replacing man in certain tasks. In 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.

Nowadays the manufacturing industry requires more man-machine interaction, meaning man and machine can work together, however accidents can happen (Guerin et al., 2019). A more efficient way to prevent accidents is to provide the machine with environment understanding, detecting the presence of the operator on the site in order to avoid dangerous movements. Therefore, the robot can calculate a new collision avoidance trajectory in real-time without slowing down nor interrupting its tasks. It is worth mentioning that the use of this practice eliminates the need to deactivate the equipment, improving production efficiency and performance control (Wisskirchen et al., 2017)

PRM has been widely used in several collision detection applications in addition to path planning (Paden et al., 2017; Baumann et al., 2010; Dias et al., 2021). The algorithm optimization of the walking path was used to reduce the radiation exposure of the staff in a radioactive environment of nuclear facility (Wang and Cai, 2018); a 3D path planning algorithm for an unmanned aerial vehicle (UAV) was developed for use in complex environments (Yan et al., 2013).

Fuzzy logic has been widely used to solve problems in control, observer stabilization, optimization, among others (Bouyahya et al., 2020; Douidi et al., 2019). In the paper from Soltanpour and Khooban (2013) an optimal fuzzy sliding mode controller was used for tracking the position of the robot manipulator, was presented. In the proposed control, initially by using the inverse dynamic method, the known sections of a robot manipulator's dynamic are eliminated.

⋆ This work was supported by the program PIBIC/PIBIT 2021/2022 funded by the IFCE/CNPq/FUNCAP.

According to the aforementioned research, it can be said that the PRM is still widely used in various applications and mainly in the path planning collision avoidance, the object of this research. This work aims to make a comparison between the implementation of PRM and PRM with Fuzzy (PRM-Fuzzy) algorithms applied to a SCARA (Selective Compliance Assembly Robot Arm) manipulator in the generation of collision-free trajectories with two static obstacles. The comparison between the two methods is demonstrated from a map, in which both algorithms are submitted to the challenge procedure of identifying the obstacles and elaborating a safe trajectory from the starting point to the q final point. The results also show the trajectories, velocities, and accelerations of the manipulator joints calculated from the inverse kinematics model.

The main contribution of this work is based on the implementation of the PRM and PRM-Fuzzy. The Fuzzy algorithm improves the PRM so that the trajectories generated by the PRM are optimized, thereby spending less time and engine power to be executed. Other contributions can be mentioned: optimized PRM with Fuzzy algorithm for collision avoidance trajectory generation; application of the PRM algorithm to a SCARA manipulator; PRM-Fuzzy algorithm for collision avoidance trajectory generation; reduction in energy consumption to carry out the trajectory because it is optimized.

2. SCARA MANIPULATOR

The SCARA (Selective Compliance Assembly Robot Arm) manipulator is a 3-DOF robot shown in Figure 1. In this work, we use only 2 DOF. The first two joints revolve around the vertical axis (z_1 and z_2) performing together parallel to the horizontal plane X_E Y_E, thus behaving as a 2-DOF planar robot.

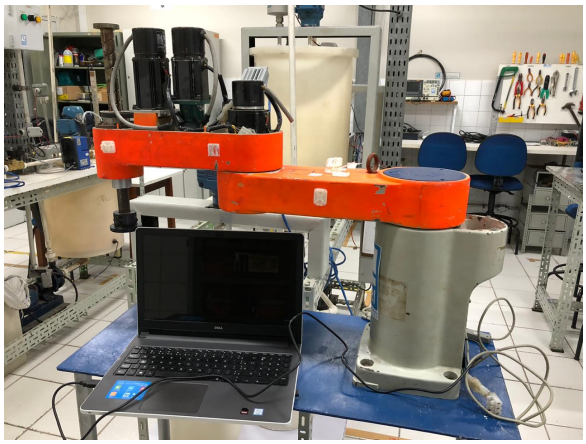

Figure 1. Robotic manipulator SCARA.

2.1 Forward Kinematics of SCARA

We use the Denavit-Hartenberg (DH) convention to develop the kinematic model and compute the parameters α, a, d and θ using the manipulator coordinate system (Hartenberg and Danavit, 1964). D-H parameters are shown in Table 1.

Table 1. DH Parameters of the SCARA manipulator.

Link	a_i	α_i	d_i	θ_i
1	0.35	0	0.32	θ_1
2	0.30	π	0	θ_2
3	0	0	d_3	0

Applying the transformation matrices and the D-H conversion, the positions in the workspace space can be achieved from the joints space coordinates

$$P_x = 0.35cos(\theta_1) + 0.30cos(\theta_1 + \theta_2) \quad (1)$$

and

$$P_y = 0.35sen(\theta_1) + 0.30sen(\theta_1 + \theta_2). \quad (2)$$

2.2 Inverse Kinematics of SCARA

From the equations of direct kinematics (1) and (2) and applying some trigonometric transformations we find the inverse kinematics equations, given by (Batista et al., 2020):

$$\theta_1 = tan^{-1}\left[\frac{P_y(L_1 + L_2cos(\theta_2)) - P_xL_2sen(\theta_2)}{P_x(L_1 + L_2cos(\theta_2)) - P_yL_2sen(\theta_2}\right] \quad (3)$$

$$\theta_2 = cos^{-1}\left(\frac{P_x^2 + P_y^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \quad (4)$$

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

2.3 Dynamics of an industrial manipulator

The dynamics of the manipulator under study will be presented using Lagrangian mechanics in the joint space. In this, dynamic equations of motion 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., 2020, 2018).

Considering the manipulator's kinetic energy, the dynamic equation can be written in a simplified way, such as:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau, \quad (5)$$

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.

2.4 Lagrange equation

Applying the Lagrange formulation and and performing all the calculations and substitutions of the terms we will have the equations that represent the dynamic model of the manipulator.

The torques of joints 1 and 2 are as follows, 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 $g = 9.8\ m/s^2$ are

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

and

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

The dynamics equations presented here are for finding the torque values that will be used to calculate the energy consumed in the trajectories of each manipulator's joint.

3. DESCRIPTION OF THE ALGORITHMS

3.1 Probabilistic Roadmap Method

The Probabilistic Roadmap (PRM) method provides motion planning to find a collision-avoidance path. It is successfully used in mobile robots in the presence of obstacles (Mohanta and Keshari, 2019). This methodology is extremely efficient, due to the ability to design paths quickly and the prevalence of the shortest path. In this method, a random sample of the configuration space is initially generated using a uniform probability distribution. In sequence, the algorithm tests the sample for collision, if it is not detected, the trajectory of the point will be generated $q_{inicial}$ to q_{final} (Sciavicco et al., 2011; Dias et al., 2021; Spong and Vidyasagar, 2020).

The logic of the PRM 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 (Sciavicco et al., 2011).

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

(1) The path is a graph $G(V, E)$;
(2) The robot configuration $q \to Q_{free}$ 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}.

3.2 Fuzzy Logic

According to Zadeh (1965), Fuzzy logic, a variable can belong to any number of sets at different levels of membership. It can be said that a Fuzzy A set is defined as the ordered pair $A = \{x,\ \mu A(x)\}$,where $x \in A$ and $0 < \mu A(x) < 1$, where $\mu A(x) = 0$, means that element x does not belong to set A and $\mu A(x) = 1$, means that the element belongs entirely to the set. The function of pertinence $\mu A(x)$ describes the degree to which object x belongs to set A. To base the Fuzzy logic, the basic operations performed with Fuzzy sets were also defined: complement, union and intersection (Nascimento Jr and Yoneyama, 2000).

In this paper a Takagi-Sugeno fuzzy (T-S) rules was utilised within the Gaussian relevance function are as follows(Takagi and Sugeno, 1985):

$$\text{IF } A_i = x \text{ AND } B = y, \text{ THEN } z = z_i(x,\ y,\ \cdots,\ r_i \quad (8)$$

where A, B are Fuzzy sets of antecedents while that the consequent is a function of the input variables.

4. METHODOLOGY

4.1 PRM with Fuzzy (PRM-Fuzzy)

As previously mentioned, in the Introduction (1), PRM is widely used in the generation of collision avoidance trajectories and Fuzzy logic is also widely used for system optimization. Here we present an improvement of the paths generated by the PRM using the Fuzzy logic, that is, PRM with Fuzzy (PRM-Fuzzy). The idea put here is that Fuzzy optimizes the paths generated by the PRM, that is, it generates a shorter path without collision. The PRM method generates collision-free paths and Fuzzy improves this path, making it smaller (optimized) and also without collision. The paths are generated in the Cartesian space. The PRM-Fuzzy algorithm is shown in Figure 2.

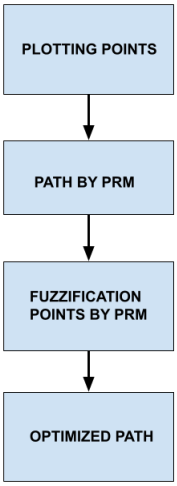

Figure 2. PRM-Fuzzy algorithm.

In the project, two main parameters were used for Fuzzy logic, spread, and rules. In order to make PRM better, 2 rules were used and spread equal to 10, for all scenarios.

4.2 Initial conditions

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 is 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 X and Y, with equal sizes whose radius is equal to 0.2 m and the side of the square is equal to 0.3 m.

The scenario is formed by the initial position of (0.0642; 0.7474), final position (0.7053; 0.0618) of the manipulator for circular obstacles, for square obstacles (0.1424; 0.7674) for beginning position e (0.7053; 0.0618) for the final position and for cross obstacles (0.1424; 0.7674) for the initial

position and (0.7053; 0.0618) for the final position, in space Cartesian. For circular obstacles, positions (0.2831; 0.6437) were used for obstacle 1 (which is at the top), (0.3135; 0.2929) for obstacle 2 (which is to the left) and (0.5474; 0.3316) for obstacle 3 (which is on the far right). For square and cross obstacles, positions (0.20; 0.60), (0.6945; 0.6) and (0.3731; 0.1022) were used in the scene, positions that define their centroids in space Cartesian.

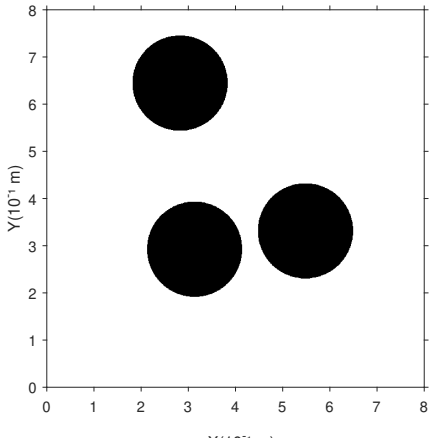

Figure 3. Map 1: circular obstacles in Cartesian space .

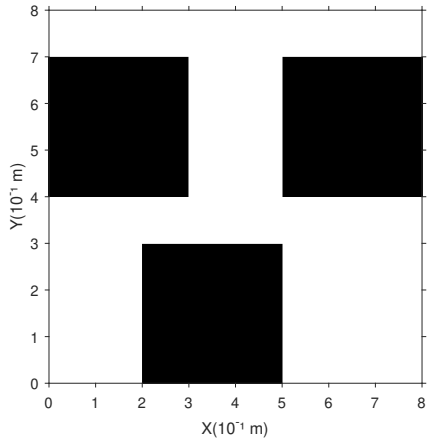

Figure 4. Map 2: square obstacles in Cartesian space .

4.3 Algorithm Comparison

Four algorithms comparison criteria were used: processing time, multiple correlation coefficient (R^2), the energy consumption of joint motors during joint displacement, and trajectory length. The average values of these criteria are calculated for 20 repetitions. The simulations were performed on a computer with a Core i7 processor - 7th Generation, with a processing speed of 2.90 GHz and RAM memory of 8.00 GB. The algorithms were implemented in m-code language with some language-specific functions.

A quantitative analysis of each algorithm in the implementation of the paths in the cartesian space is given by the

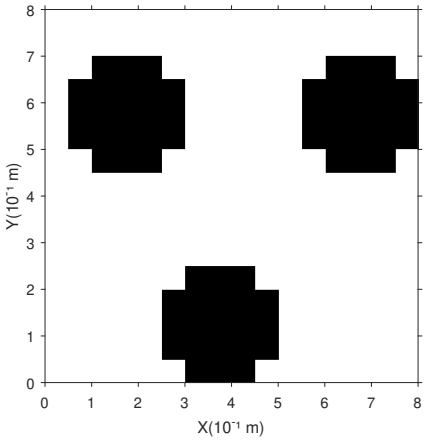

Figure 5. Map 3: cross-shaped obstacles in Cartesian space.

performance indexes: multiple correlation coefficient, (R^2) of each algorithm. The Eq. (9) presents R^2 given by,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{9}$$

where y_i are the observed data, $\bar{y}$ is the mean of the observed data, and $\hat{y}$ data estimated by the model.

For each test performed, the mechanical energy consumption and the motors torque of joints 1 and 2 were calculated. To assist in the execution of the motor torque calculation, the motors power information was necessary.

To calculate the torque of motors 1 and 2, the following equation was used

$$T_m = 9,55\frac{P_m}{N_m} \tag{10}$$

where, T_m is the torque of motors in Nm, P_m is the nominal power of motors in W and N_m the value of the motors speed in rpm (min^{-1}).

Mechanical energy was calculated from:

$$E = \sum_{i=1}^{n}(P_{mi}t_i) \tag{11}$$

where E is the motor's mechanical energy in kWh, P is the power calculated during the tests in W and t is the time of the tests in h (hours).

5. RESULTS

In this section, we present the results and discussion of the performance comparison between the PRM and PRM-Fuzzy algorithms for different conditions, with obstacles circular, square, and cross-shaped. The trajectories, velocities, accelerations, and torques of the joints are shown comparing the collision-free paths reached.

5.1 Paths in Cartesian space

Here are presented the results of the paths generated by the PRM and PRM-Fuzzy algorithms for three different scenario types with obstacles three static obstacles with circular, square, and cross-shaped (as shown in 4.2).

The generated paths are in the Cartesian space for each scenario. Figure 6 shows the PRM algorithm and PRM-Fuzzy for the circular obstacles, Figure 7 for the square obstacles, and Fig 8 for the cross-shaped obstacles. In all figures, the path in the color green, collision-free, is the path generated for PRM-Fuzzy, the other path is generated by conventional PRM.

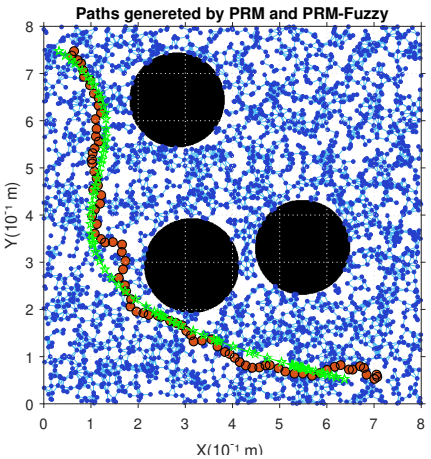

Figure 6. Paths in the Cartesian space of the PRM algorithm and PRM-Fuzzy for circular obstacles.

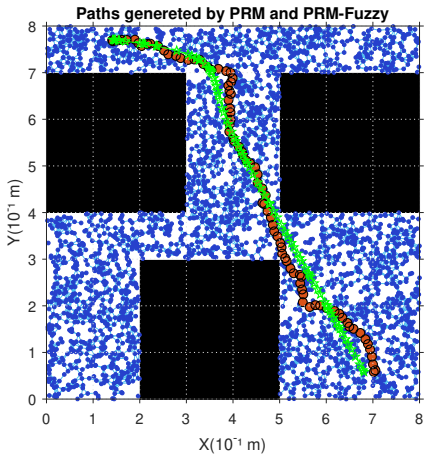

Figure 7. Paths in the Cartesian space of the PRM algorithm and PRM-Fuzzy for the square obstacles.

5.2 Trajectories in the joint space

The following are presented the trajectories, velocities, accelerations, and torques of joints 1 and 2 of the manipulator for circular, square and cross-shaped obstacles. The trajectories in the space of the joints were obtained from the resolution of the inverse kinematics (Equations (3) and (4)) and the paths generated by PRM and PRM-Fuzzy presented in figures 6, 7, and 8 and as the input the points of the trajectory in the Cartesian space.

Figures 9, 10, and 11 presented the trajectories, velocities, and accelerations of joints 1 and 2 of the manipulator, respectively, generated by PRM and PRM-Fuzzy for circular obstacles.

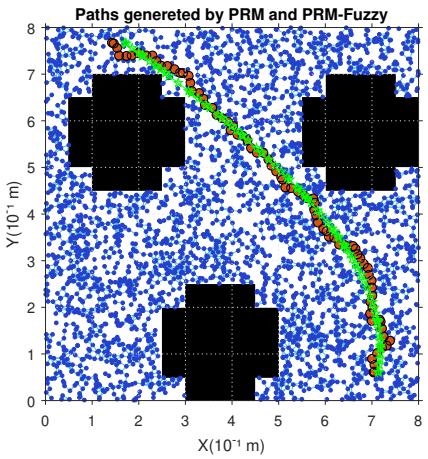

Figure 8. Paths in the Cartesian space of the PRM algorithm and PRM-Fuzzy for the cross-shaped obstacles.

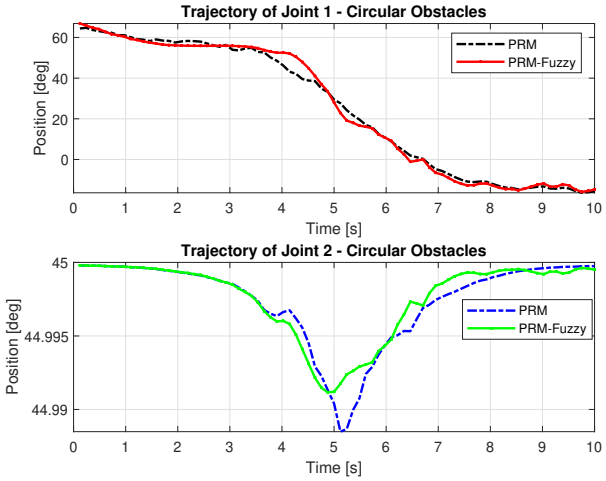

Figure 9. SCARA manipulator joint trajectories generated from PRM and PRM-Fuzzy for circular obstacles.

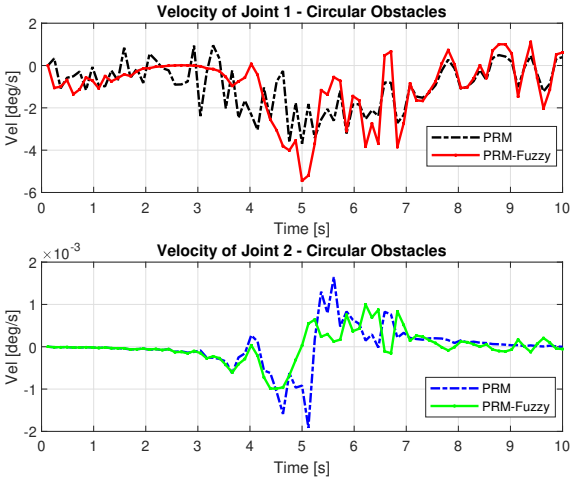

Figure 10. SCARA manipulator joint velocities generated from PRM and PRM-Fuzzy for circular obstacles.

The joint torques were obtained from the dynamic model, Equations (6) and (7) of the manipulator and are shown

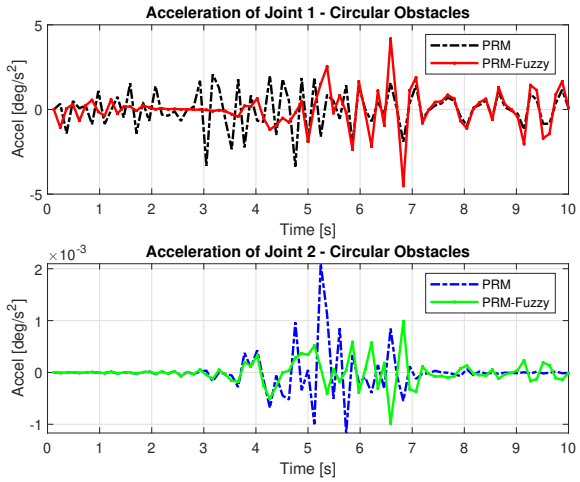

Figure 11. SCARA manipulator joint accelerations generated from PRM and PRM-Fuzzy for circular obstacles.

in Figure 12. Torques were calculated by taking the trajectories, velocities, and accelerations shown in Figures 9, 10, and 11.

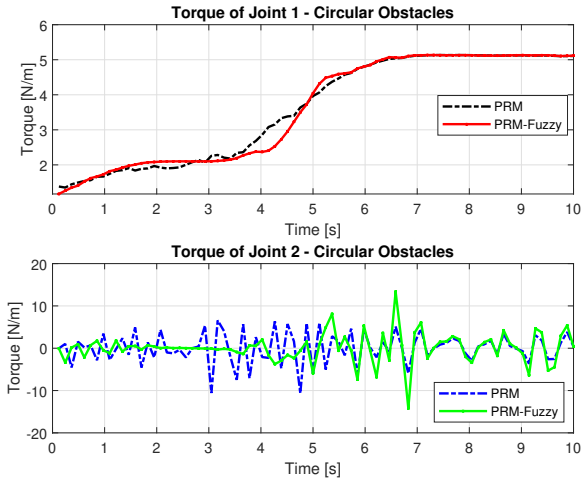

Figure 12. SCARA manipulator joint torques calculated from trajectories, velocities, and accelerations from PRM and PRM-Fuzzy for circular obstacles.

Figures 13, 14, and 15 presented the trajectories, velocities, and accelerations of joints 1 and 2 of the manipulator, respectively, generated by PRM and PRM-Fuzzy for square obstacles.

The joint torques were obtained from the dynamic model, Equations (6) and (7) of the manipulator and are shown in Figure 16. Torques were calculated by taking the trajectories, velocities, and accelerations shown in Figures 13, 14, and 15.

Figures 17, 18, and 19 presented the trajectories, velocities, and accelerations of joints 1 and 2 of the manipulator, respectively, generated by PRM and PRM-Fuzzy for square obstacles.

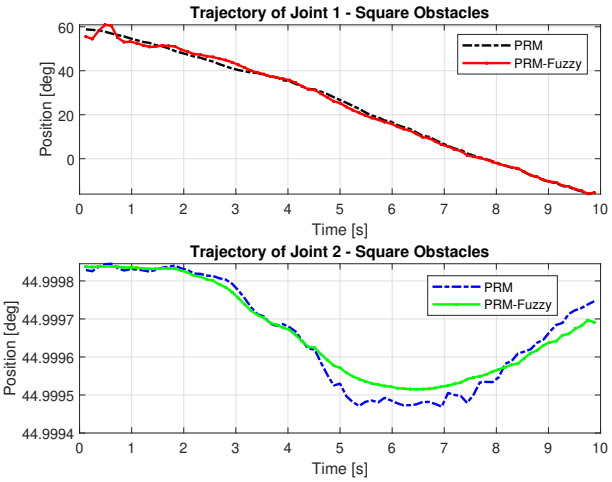

Figure 13. SCARA manipulator joint trajectories generated from PRM and PRM-Fuzzy for square obstacles.

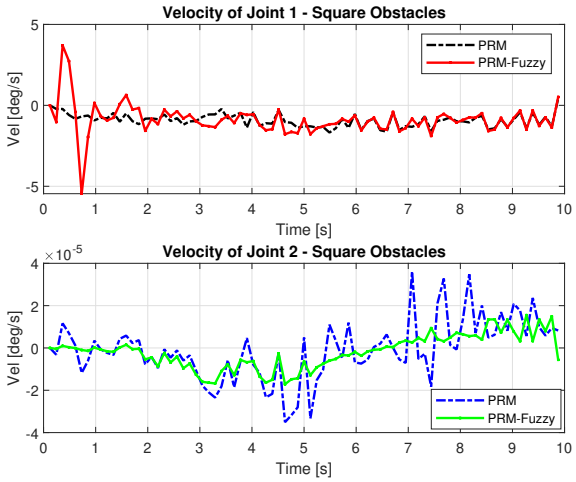

Figure 14. SCARA manipulator joint velocities generated from PRM and PRM-Fuzzy for square obstacles.

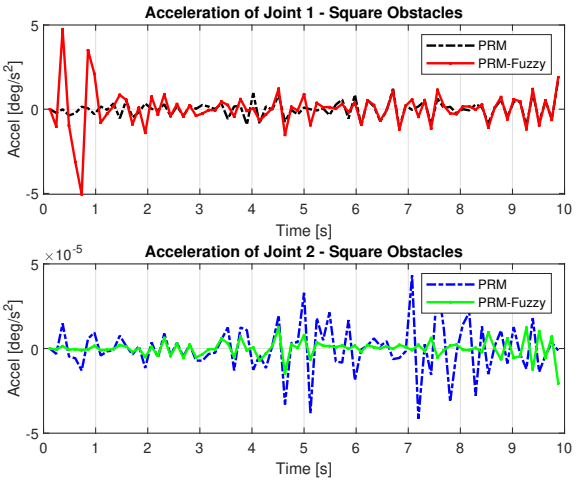

Figure 15. SCARA manipulator joint accelerations generated from PRM and PRM-Fuzzy for square obstacles.

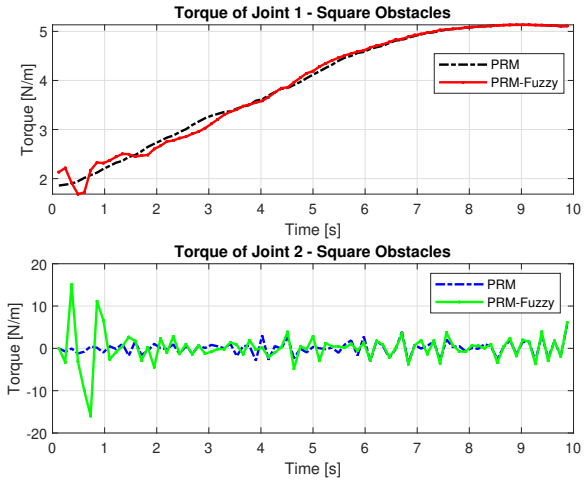

Figure 16. SCARA manipulator joint torques calculated from trajectories, velocities, and accelerations from PRM and PRM-Fuzzy for square obstacles.

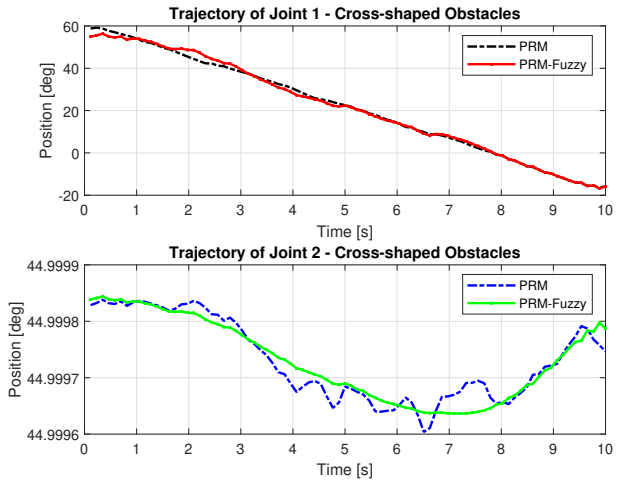

Figure 17. SCARA manipulator joint trajectories generated from PRM and PRM-Fuzzy for cross-shaped obstacles.

The joint torques were obtained from the dynamic model, Equations (6) and (7) of the manipulator and are shown in Figure 20. Torques were calculated by taking the trajectories, velocities, and accelerations shown in Figures 17, 18, and 19.

5.3 Discussions

For a comparison of the algorithms, a were calculated average processing time, multiple correlation coefficient (R^2), the energy mechanical consumption of joint motors during joint displacement, and trajectory length for all scenarios with circular, square, and cross-shaped obstacles. The values of these criteria for each type of obstacle are presented in Tables 2, 3 and 4, respectively.

In all scenarios presented, the PRIM-Fuzzy algorithm presented better results than the conventional PRM (as highlighted in the Tables 2, 3 and 4, for circular, square, and cross-shaped obstacles, respectively.

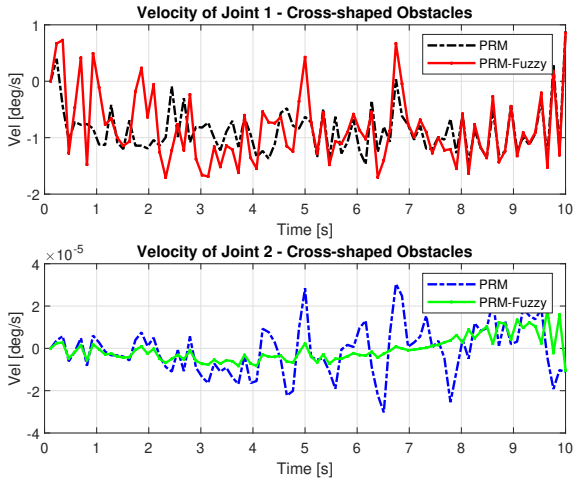

Figure 18. SCARA manipulator joint velocities generated from PRM and PRM-Fuzzy for cross-shaped obstacles.

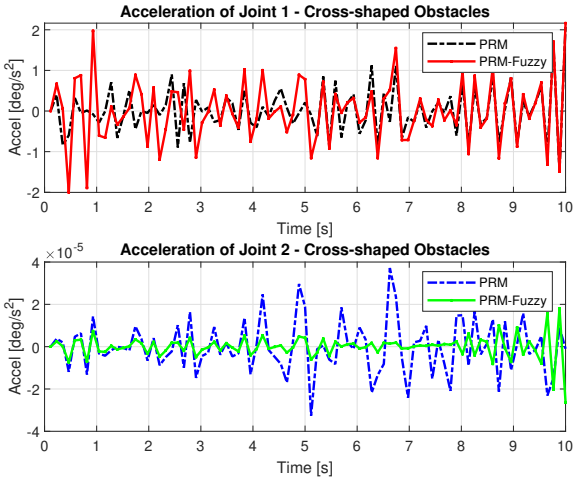

Figure 19. SCARA manipulator joint accelerations generated from PRM and PRM-Fuzzy for cross-shaped obstacles.

Table 2. Comparison of methods for circular obstacles.

Method	Time [s]	R^2	Energy Cons. [kWh]	Traj. [m]
PRM	5.2980	0.9799	0.07498	1.4094
PRM-Fuzzy	4.3824	0.9178	**0.05334**	**1.2583**

Table 3. Comparison of methods for square obstacles.

Method	Time [s]	R^2	Energy Cons. [kWh]	Traj. [m]
PRM	4.3824	0.9738	0.0711	1.0938
PRM-Fuzzy	4.2321	0.9943	**0.0670**	**1.0415**

Table 4. Comparison of methods for cross-shaped obstacles.

Method	Time [s]	R^2	Energy Cons. [kWh]	Traj. [m]
PRM	4.0341	0.9497	0.0715	1.1609
PRM-Fuzzy	3.9824	0.9873	**0.0705**	**1.0344**

It's important to highlight that in all scenarios the PRM-Fuzzy was better than the conventional PRM for the pro-

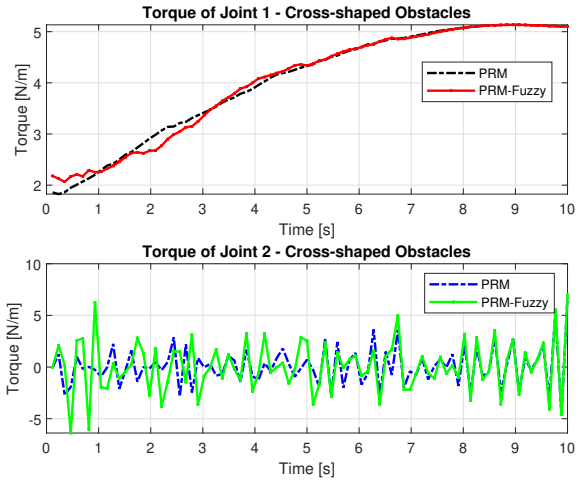

Figure 20. SCARA manipulator joint torques calculated from trajectories, velocities, and accelerations from PRM and PRM-Fuzzy for cross-shaped obstacles.

posed application. Two points that stand out are energy consumption and trajectory length, to move the manipulator from the start point to the endpoint. This comparison is important nowadays, as there is great concern in the fact that we consume less energy, and this point is of fundamental importance in all fields.

6. CONCLUSION

This paper presented an alternative improvement of the PRM algorithm using Fuzzy logic, PRM-Fuzzy, for the generation of trajectory with collision avoidance applied to a SCARA manipulator. The results were shown to be consistent and satisfactory in the collision-free trajectory generation using the algorithm PRM-Fuzzy according to the comparisons. The PRM-Fuzzy showed better efficiency than the PRM, in the criteria tested: average processing time, multiple correlation coefficient (R^2), the energy mechanical energy consumption of joint motors during joint displacement, and trajectory length for all scenarios with circular, square, and cross-shaped obstacles.

It can also be concluded that this paper makes a contribution to the area of path planning collision avoidance by presenting an improvement to the PRM algorithm. This work is also a foundation for collaborative robotics that is a pillar of industry 4.0. The authors continue to research in this field to further improve collision avoidance algorithms to apply to affine manipulators for collaborative robotics applications.

REFERENCES

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

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

Baumann, M., Leonard, S., Croft, E.A., and Little, J.J. (2010). Path planning for improved visibility using a probabilistic road map. *IEEE Transactions on Robotics*, 26(1), 195–200.

Bouyahya, A., Manai, Y., and Haggège, J. (2020). Fuzzy observer stabilization for discrete-time takagi–sugeno uncertain systems with k-samples variations. *Journal of Control, Automation and Electrical Systems*, 31(3), 574–587.

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.

Douidi, B., Mokrani, L., and Machmoum, M. (2019). A new cascade fuzzy power system stabilizer for multi-machine system stability enhancement. *Journal of Control, Automation and Electrical Systems*, 30(5), 765–779.

Guerin, C., Rauffet, P., Chauvin, C., and Martin, E. (2019). Toward production operator 4.0: modelling human-machine cooperation in industry 4.0 with cognitive work analysis. *IFAC-PapersOnLine*, 52(19), 73–78.

Hartenberg, R. and Danavit, J. (1964). *Kinematic synthesis of linkages*. New York: McGraw-Hill.

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

Nascimento Jr, C.L. and Yoneyama, T. (2000). Inteligência artificial em controle e automação. *Editora Edgard Blücher Ltda.*

Paden, B., Nager, Y., and Frazzoli, E. (2017). Landmark guided probabilistic roadmap queries. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4828–4834. IEEE.

Sciavicco, L., Siciliano, B., Villani, L., and Oriolo, G. (2011). Robotics: Modelling, planning and control, ser. advanced textbooks in control and signal processing.

Soltanpour, M.R. and Khooban, M.H. (2013). A particle swarm optimization approach for fuzzy sliding mode control for tracking the robot manipulator. *Nonlinear Dynamics*, 74(1), 467–478.

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

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, 1, 116–132.

Wang, Z. and Cai, J. (2018). Probabilistic roadmap method for path-planning in radioactive environment of nuclear facilities. *Progress in Nuclear Energy*, 109, 113–120.

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.

Yan, F., Liu, Y.S., and Xiao, J.Z. (2013). Path planning in complex 3d environments using a probabilistic roadmap method. *International Journal of Automation and computing*, 10(6), 525–533.

Zadeh, L.A. (1965). Information and control. *Fuzzy sets*, 8(3), 338–353.