23rd ABCM International Congress of Mechanical Engineering
December 6-11, 2015, Rio de Janeiro, RJ, Brazil

PATH PLANNING APPLIED TO A CYLINDRICAL MANIPULATOR CONSIDERING DEVIATION OBSTACLES BASED ON FIREFLY METAHEURISTIC ALGORITHM (FMA)

Rafael Crespo Izquierdo
Anselmo Rafael Cukla
Flávio José Lorini
Eduardo André Perondi
Federal University of Rio Grande do Sul (UFRGS) – Department of Mechanical Engineering (DEMEC), Sarmento Leite St, 425, Downtown, 90050-170 – Porto Alegre, RS – Brazil.
rcrespo9@hotmail.com, cukla.anselmo@gmail.com, lorini@ufrgs.br, perondi@mecanica.ufrgs.br.

Leonardo Missiaggia
Sul-Rio-Grandense Federal Institute of Education, Science and Technology (IFSul) – Campus Camaquã, Ana Gonçalves da Silva St., 901, Olaria, 96180-000 – Camaquã, RS – Brazil.
missiaggia@camaqua.ifsul.edu.br[5]

Abstract. *Robot manipulators path planning consists in an important issue that affects significantly the performance of automatic industrial process. By using an only suitable trajectory it is possible to meet requirements such as reduced cycle times, obstacles prevention in the robot workspace and smooth movements performed by handling parts. Several studies related to trajectory curvatures, time optimization, jerk, and energy minimization, among others, may be found in related literature. In this study, we use an optimization technique based on Metaheuristic Algorithm Firefly (FMA) in order to do the trajectory planning of a robot manipulator with 5 degrees of freedom, also taking into account the obstacle avoidance problem. The strategy consists in identifying the intermediate point of the path by means of FMA algorithm, executing a posterior interpolation through a polynomial function to obtain the trajectory to be executed by the robot. In a practical application, it is necessary to define an objective function for the solution of the problem, for example, minimizing the travelled distance by the manipulator arm, considering restrictions associated to the eventual existence of obstacles. Based on results analysis it was verified that this methodology meets the performance goals of metaheuristic optimization techniques to get an adequate planning of robot trajectories.*

Keywords: *Robotics, Trajectory Planning, Metaheuristic Algorithms*

1. INTRODUCTION

In recent times, industrial manipulators have been used in various applications, such as risky, repetitive, or unhealthy tasks. They are also used in situations in which they could replace man activities with high advantages, automating and standardizing processes.

These applications require that robots interact in environments with different types of objects, such as, motors, belts, boxes, pallets, among others. In order to meet these applications, robots must be able to perform their tasks with specially planned path, suitable for these peculiar environments, also to guarantee good operating performance.

For a satisfactory performance of robot control, there must be a reference path in consonance with robot features (topology, control algorithms and application, among others). Improper path would possibly lead the handler to perform abrupt movements, which can compromise their dynamic performance and even physical integrity of its components.

This study presents a strategy for robotic manipulators path planning that provides short time paths in face of static obstacles. The optimization strategy is based on the metaheuristic method whose algorithm is based on fireflies behavior (Yang, 2010).

The literature on the design of robotic manipulators paths using stochastic optimization algorithms considering the obstacle avoidance is relatively short, although some studies constitute relevant scientific papers, for instance, Severin and Rossmann (2012) which compared the performance of three metaheuristic algorithms: Particle Swarm Optimization (PSO), Firefly (FMA) and Harmony Search (HS). Severin and Rossmann proposed paths be generated through a sequence of points obtained via these three algorithms. To generate these points, one should apply an objective function which evaluates the overall movement of the robot joints and the time required to traverse these points, proving details of point-to-point interpolation strategies in industrial robots. Other important contributions to the field are described and commented next.

Baba and Kubota (1994) used genetic algorithms (GA) to determine the path taken by the robot between a given start and end point. Under this method, robot workspace is represented by cells with a Potential Value (PV) predefined. The VP is used to prevent possible collisions between the robot and other movable barriers, and therefore to calculate the objective function for this risk. The results reveal that the proposed method is effective since it addresses the problem of collision avoidance in different sandboxes.

Tian and Collins (2004) propose planning paths with obstacles for a robot with two degrees of freedom. The method consists of two steps. Initially, the AG is used for generating intermediate path points, and then those points are interpolated by a cubic Hermitian function. Results demonstrate that such method has the following advantages: easier programming, efficient implementation of a smooth path and it could even be extended to spatial and redundancy problems in robots.

Chyan and Ponnambalam (2012) feature four variations of the PSO algorithm to plan the trajectory of a robot with 5 degrees of freedom. This approach consists in moving the robot, which is represented in a two-dimensional plane, from an initial position to a final position with minimum error, avoiding collision with static obstacles. To implement these algorithms, one applies an objective function that evaluates the existing error between the current position and the end point (target).

Regarding the study associated with the shape of paths or path interpolation techniques, several authors may be quoted. Lin et al (1983), for instance, addresses the problem of control for an industrial robot through the optimization technique, based on optimal control. This approach is applied in two steps: initially carrying out offline path planning, and then online controlling the path. The course is designed in robot joint space, using splines of 3rd order interpolation of the points. The points (nodes) of the path are set so that velocities, accelerations and jerk be limited, and travel time be minimized within the possible functions which are point-to-point calculated.

Tangpattanakul and Artrit (2009), use the HS algorithm to minimize time path of a robot manipulator of 6 GDL through cubic splines. The proposed algorithm considers that the minimal time of a path be defined as objective function of the problem so that the speeds, acceleration and jerk be established as an optimization problem of restrictions. The results show that the proposed method guarantees continuity of functions in terms of position, velocity and acceleration.

This paper proposes a path planning strategy that consists in generating waypoints with subsequent interpolation. The generated midpoints are based on the methodology presented by Severin and Rossmann (2012). In order to interpolate these points, one applies the optimization of path interpolation strategy as presented by Simon (2004) and Missiaggia (2014). Via case study, the strategy is applied to path planning of a cylindrical robot 5 GDL (degrees of freedom) with pneumatic drive, now called CPR. This robot is under development at the Laboratory of Mechatronics and Control at Federal University of Rio Grande do Sul (LAMECC/UFRGS) and its main application consists of parts handling tasks.

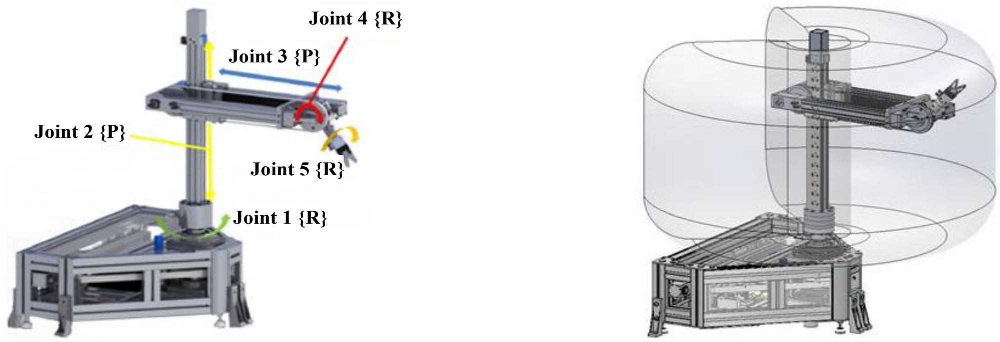
The study is here organized as follows: Section 2 presents features of the manipulator robot, the method applied to identify obstacles in the volume of robot labor and the working principle of Metaheuristic Firefly Algorithm (FMA). Section 3 addresses the study on the robot path planning, which outlines the methodology for generating the waypoints of the path and the problem formulation of these interpolation points. Section 4 presents and discusses findings of a case study, which brings up results related to obstacle-in-the-path prevention and jerk values from the interpolated path. Section 5 presents the conclusions of this work.

2. PROBLEM DESCRIPTION

As previously mentioned, this paper proposes a method to path planning to be tested in a pneumatic 5-degree-of-freedom robot (CPR). Robot's mobile arm moves within a working space where there may be static obstacles. The generated paths should be optimized according to certain objective functions to provide an adequate benchmark for the robot control algorithm. As a consequence, the path synthesis method proposed in this study shall, for clarity of the issues here discussed and without loss of generality, be filed concurrently with the description of its application to CPR.

2.1 Features of the pneumatic robot

Research and applications in Robotics, including details on the development of RPC controller, can be found in works such as Allgayer (2011), Rijo (2013), Missiaggia (2014) and Sarmanho Jr. (2014). Configuration joints of RPP: RR form the handler presented in Figure 1, which depicts the handler structure indicating the disposition of joints (a), and workload (b).



(a) Joint configuration **(b) Representation of cylindrical workspace**

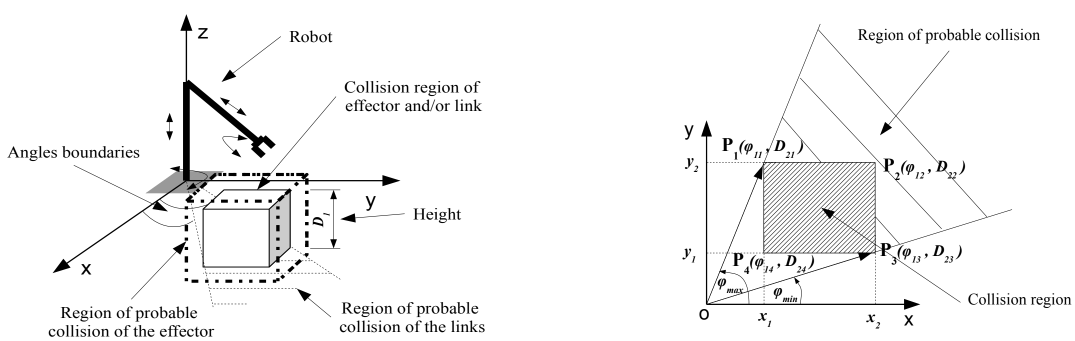
Figure 1. Features of RPC (adapted from Missiaggia, 2014)

With respect to the RPC, Missiaggia (2014) was the first research applied to path planning. The contribution of the present paper is the development of a strategy for generating optimized paths for the RPC. From Missiaggia's method application altogether with some optimization criteria, it is possible to obtain joint area paths that result in the expected movement of the robot end-effectors. Obtaining paths within joint areas from an expected path in the general working space is possible by means of an algebraic solution of inverse kinematics; and for generating intermediate paths in the robot workspace, an interpolation algorithm based on seventh degree polynomial splines is used. This choice ensures the continuity of function position and its three first derivatives, this being a necessary condition for the implementation of relevant strategies based on control by calculated torque (Slotine and Li, 1988). The proposed method for the generation of splines, based on Simon's work (2004), allows curves in joint space with optimized values of jerk, acceleration or velocity, by adjusting parameters depending on the requirements of each application.

2.2 Mapping of obstacles in the volume of the manipulator work

For purposes of simplification, the case study considers the situation where there is only a single static obstacle in the robot workload. It is also defined that the volume surrounding the object is represented by a cube and that the information concerning the obstacle is provided by sensors which are integrated into computer vision systems.

Mathematically, the collision zones are calculated through limited areas in the robot workload. As shown in Figure 2, there are two defined regions, and the region (or space) of the collision object is directly related to the volume, whereas the possible collision area is associated with the scale and geometry of the robot.



(a) xy plane **(b) xz plane**

Figure 2. Restriction areas in the robot workspace

The critical area is determined by angular variation (φmax and φmin) of robot's first joint and object height D1, as shown in Figure 2. The collision area is the volume represented by points P1, P2, P3 and P4 and D1 height. The possible collision region is defined as the handler difficult access area and is further the object relative to the origin of robot's coordinate system, as shown in Figure 2b. This approach also considers a possible collision area which is defined at a distance of 0.1m around collision region.

Collision with the obstacle can occur in three situations:

i) effector collision: this collision occurs when any point on the path exceeds the boundaries of a safety area defined by a given distance from collision region.

ii) Robot links Collision: it occurs when robot links surpass the collision area.

iii) Simultaneous collision of effector and links: it only occurs when the path points are located within the collision area, being the point of a given path within the object volume.

According to Severin and Rossmann (2012), automatic generation of direct paths (point to point) between two location points of the effector of a robot manipulator can lead to collision between the robot and possible obstacles. In accordance with the same authors, it is necessary to define intermediate positions when generating a desired path in order to avoid possible collisions. This procedure is also adopted in the present work.

2.3 Firefly Metaheuristic Algorithm (FMA)

Firefly Metaheuristic Algorithm (FMA) was developed by Yang , 2007, based on observations of brightness behavior (bioluminescence) of fireflies. According to Yang , 2010, the optimization process via FMA algorithm is based on three ideal rules: 1) all fireflies are considered unisex , with the intention that a firefly is attracted to other fireflies . 2) The attraction is related in proportion to its brightness. It means that independently of fireflies, the dimmer will move to the brightest. The attraction is proportional to the brightness and both diminish in extent as its distance increases. 3) The brightness of a firefly is associated with an indication of the objective function.

As the attractiveness of a firefly is proportional to its luminosity perceived by other adjacent fireflies, one can set the β attractiveness as a function of distance r as follows:

$$\beta = \beta_0 e^{-\gamma r^2}, \tag{1}$$

where β0 is the attraction at r = 0. The distance r_{ij} between two fireflies is defined as its Cartesian distance. The movement of firefly i who is attracted to other more attractive firefly j (brighter) is determined by Eq. (2):

$$x^{i+1} = \beta_0 e^{-\gamma r_{ij}^2} + \left(x^i - x^j\right)\alpha\varepsilon^i, \tag{2}$$

being xi and xj the current positions of fireflies and x^{i+1} the ith position of next generation firefly. Analyzing Eq (2), the following features are defined: the equation second term is related to attraction, while its third term is their randomization. The term α is randomization parameter while ε^i is defined as a vector of random numbers resulting from a Gaussian or uniform distribution.

3. PATH PLANNING STUDY

Robot path planning is presented in two stages. The first one consists in planning the intermediate points of the path, which are obtained through FMA, adapted from Severin and Rossmann, 2012. The second stage is summarized in the interpolation between these points by splines, seeking a path that meets robot control requirements.

3.1 Waypoints Path Planning

The solution vector, which is the design variable of an optimization problem for a kinematic device with n degrees of freedom and p intermediate position is defined as follows:

$$sol = \left(\left(g_{1,1}, g_{2,1}, \ldots\ldots, g_{n,1}\right), \ldots\ldots, \left(g_{1,p}, g_{2,p}, \ldots\ldots, g_{n,p}\right)\right), \tag{3}$$

where gi , j is the value of the joint j at position i . The performance of a given solution depends on the following situations : the movement must be free of collisions , the movement time required should be minimal and there should be no unnecessary movements .

where $g_{i,j}$ is the joint j value at position i . The performance of a given solution depends on the following situations: the movement must be free of collisions, the required movement time should be minimal and there should be no unnecessary movements.

The objective function that describes this problem is defined as seen in (4), (5) and (6):

$$F_i = t_{sol} + \frac{w_{sol}}{w_{\max}} IPO \, , \tag{4}$$

$$F_i = 2t_{\max} - \left(t_{sol} + \frac{w_{sol}}{w_{\max}} IPO \right), \tag{5}$$

$$F_i = 2t_{\max} - \left(t_{sol} + \frac{w_{sol}}{w_{\max}} IPO \right), \tag{6}$$

where t_{sol} is defined as the time for a specific solution, IPO is the interpolation interval to define all-the-way points, t_{max} is the time for the longest possible solution, w_{sol} is accumulated motion of all joints to a specific solution and w_{max} is the movement of all axes for a given path.

$$w_{sol} = \sum_N \sum_I \left| g_{n,i} - g_{n,i-1} \right| \tag{7}$$

where N is the number of joints, I is the number of interpolation steps, and g_{n-1} is the value of joint vector of nth element for interpolation in step i.

As seen in the equations related to objective function calculation, there are different criteria to apply equations in optimization problem. The several representations for objective function math consist in the relationship between the points which are generated by the algorithm and the collision regions presented in Section 2.2.

In Eq.(4) one shall consider that there are no collisions between the track formed by randomly generated points (via algorithm) and the collision region. Eq.(5) is used when collisions occur on the robot end-effector, requiring lengthen the route to find a way around the obstacle. Eq.(6) is used in cases where the collision occurs to robot shoulder, when a shortest path could prevent the occurrence of collisions.

3.2 Path planning considering midways

Once the path waypoints are known, one must define a function to interpolate these points. In this paper paths were created via eighth-order splines, as proposed by Simon, 2004, ensuring continuous paths in speed, acceleration and jerk in robot joints. For any order splines paths, it is initially assumed that its nodes are equally spaced, so that ti = (i = 0, ..., N), where N is the number of polynomial segments of the path and M is a positive integer. Simon, 2004 defines a M-1 degree spline as a function y(t) that satisfies the criterion corresponding to the equations (8) to (10):

$$y(t) = y_i(t), \qquad t \in [t_{i-1}, t_i], \qquad (i = 1, \ldots\ldots, N) \tag{8}$$

$$y_i(t) = x_i(t - t_{i-1}), \tag{9}$$

$$x_i(t) = \sum_{k=0}^{M-1} a_{ik} t^k, \qquad t \in [0, T], \tag{10}$$

being T the normalized length of each function segment. The spline y (t) is the union of N continuous segments y_i (t) (i = 1, ..., N). Each x_i (t) is a normalized spline segment from t = 0 to t = T. These normalized segments are then displaced in time to form the segments y_i (t), which are joined to form spline function y (t) of length NT. The a_{ik} parameters (k = 0, ..., M-1) are the coefficients of the ith-segment spline. The final function y (t) and its derivatives y '(t), ..., y (R) (t) are continuous between t_0 and t_N, provided that R = (M/2)-1, R being the number derivatives of y(t), which are able to be made continuously. Simon, 2004 also proposes a function that allows adjustment of resulting spline features considering specific application and held parameterization. To do so, one shall determine, among all continuous functions that satisfy the criteria of equations (8) to (10), which one minimizes the value of J function, expressed by Eq. (11):

$$J = \sum_{I=0}^{N} p_i (y_i - f_i)^2 + \int_{t_0}^{t_N} \left\{ \alpha_1 [y'(t)]^2 + \ldots. + \alpha_R [y^{(R)}(t)]^2 \right\} dt \tag{11}$$

In Eq. (11), $y_i \equiv y(t_i)$ and pi values allow monitoring the curve fit for each node, while the parameters α_i enable control of the relative magnitude of each function derivative. In practice, the control parameters p_i are usually constant for any data range, and so it is the spacing between the feature points considered to be fixed, so that ti-ti-1 = h.

In the applied algorithm, it is required to continue the function y (t) and their first R which are derived in all intermediate function nodes. Therefore, considering $y_i \equiv y(t_i)$, $y'_i \equiv y'(t_i)$, $y_i^{(R)} \equiv y^{(R)}(t_i)$, one can get from the criteria specified by equations (8) to (10), M, or 8 constraint equations of the form:

$$y_{i-1}^{(r)} = x_i^{(r)}(0) \qquad (r = 0, \ldots\ldots, R), \tag{12}$$

$$y_i^{(r)} = x_i^{(r)}(T) \qquad (r = 0, \ldots\ldots, R). \tag{13}$$

The specified points to form each joint path are approximated by splines which are formed from seventh-degree polynomial segments, ensuring that up to third order derivatives are continuous (R = 3). Whereas T = 1 is the normalized length of between-node segments, by equations (12) and (13) it is possible to determine eight constraints (written in matrix form) resulting in Eq. (14). Each standard spline segment has therefore 8 coefficients (M = 8). Considering N segments in the N polynomial spline function, there are 8N independent variables to be determined. The optimization of cost function is considered an optimization with regard to these 8N coefficients, subject to the continuity constraints of y (t), y (t), y '(t) and y' "(T) nodes; and since the coefficients are functions y_i, y'_i, y''_i e y'''_i (i = 1, ..., N), the problem can be reformulated as an optimization without constraints concerning 4 (N + 1) parameters. Thus the minimization of cost function can be resolved through Eq. (15).

$$\begin{bmatrix} y_{i-1} \\ y_i \\ y'_{i-1} \\ y'_i \\ y''_{i-1} \\ y''_i \\ y'''_{i-1} \\ y'''_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6 & 12 & 20 & 30 & 42 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 24 & 60 & 120 & 210 \end{bmatrix} \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \\ a_{i6} \\ a_{i7} \end{bmatrix} \qquad (i = 1, \ldots\ldots, N), \tag{14}$$

$$\frac{\partial J}{\partial y_i^{(r)}} = 0, \qquad (i = 1, \ldots\ldots, N), \quad (r = 0, \ldots\ldots, 3). \tag{15}$$

Rewriting Eq. (11) as a sum of smaller integral equations and representing the spacing between the nodes as a fixed hour period h, r derived is multiplied by a factor (T/h)'. Thus:

$$J = \sum_{i=0}^{N} p_i (y_i - f_i)^2 + \sum_{k=1}^{N} \int_0^T \left\{ \left(\frac{T}{h}\right) \alpha_1 [y_k'(t)]^2 + \left(\frac{T}{h}\right)^3 \alpha_2 [y_k''(t)]^2 + \left(\frac{T}{h}\right)^5 \alpha_3 [y_k'''(t)]^2 \right\} dt \tag{16}$$

Assuming that $y_k(t)$ is a function $y_i^{(r)}$ only for k = i and k = i + 1, and combining equations (15) and (16), one can rewrite the criterion of minimization by applying equations (17) to (19):

$$2 p_0 (y_0 - f_0) \delta_r + \frac{\partial J_1}{\partial y_0^{(r)}} = 0, \qquad (r = 0, \ldots\ldots, 3); \tag{17}$$

$$2 p_i (y_i - f_i) \delta_r + \frac{\partial (J_i + J_{i+1})}{\partial y_0^{(r)}} = 0, \qquad (r = 0, \ldots\ldots, 3); \quad (i = 1, \ldots\ldots, N-1) \tag{18}$$

$$2 p_N (y_N - f_N) \delta_r + \frac{\partial J_N}{\partial y_N^{(r)}} = 0, \qquad (r = 0, \ldots\ldots, 3); \tag{19}$$

where $\delta_r = 1$ if $r = 0$ and $\delta_r = 0$ if $r \neq 0$. Using a math software of symbolic manipulation, equations (17) to (19) were solved, resulting in 4 (N + 1) equations for a total of 4 (N + 1) of y_i, y'_i, y''_i and y'''_i values to be determined. Once they are obtained, these values are replaced into Eq. (14), determining optimal coefficients for spline according to the criteria defined by weighing pi and ai coefficients.

4. RESULTS

In the proposed case study, the effector RPC runs a set of 5 points allocated in robot workspace. The key points for path generation were obtained by FMA optimization algorithm, while the start and end points have been arbitrarily set. To illustrate the problem, the following restrictions are considered: 1) The start and end points are p 0 = (0.5; 00:45; 12:25) and p5 = (-0.5; 12:45; 12:25), respectively; 2) Θ4 and Θ5 angles are kept constant along the path; 3) the obstacle is static and located within working volume, with predefined size and position; 4) the course must be covered in a time of 4 seconds and present maximum deviation value of 20 uM in the intermediate nodes. Knowing the path execution time and the number of segments between nodes (N = 4) one can determine the value for the parameter h, which results in (h = 4/4 = 1s). Once determined the value of parameter h, simulations may be performed by varying the value of p in order to find the path that satisfies the first specified condition.

Figure 3 presents the path nodes (obtained from FMA) and interpolation results from within joint areas.

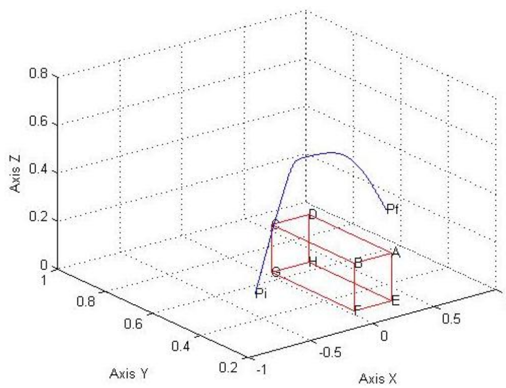


Figure 3. Three-dimensional path after joint interpolation process

One can see that the path obtained by interpolation within joint area results in a curve that meets the requirement of not clashing with the proposed obstacle because the path points are not in the critical regions presented in item 2.2. Regarding the implementation of FMA, it is observed that the proposed method can be applied in free-from-collision path planning for this robot. However, it is important to emphasize that not always the best solution obtained by the algorithm ensures that a collision can occur. The optimal parameters of FMA to the problem were the following: n = 50, α = 1, β0 = 0.5, the maximum number of iterations was 50 and the algorithm processing time was 53 seconds. Considering the example from Severin and Rassmann, 2012, the time to go through the waypoints by a point-by-point path was 3.98 s, being this the value of objective function in this case. Figure 4 shows the waypoints obtained by FMA and comparative jerk curves in joint area.

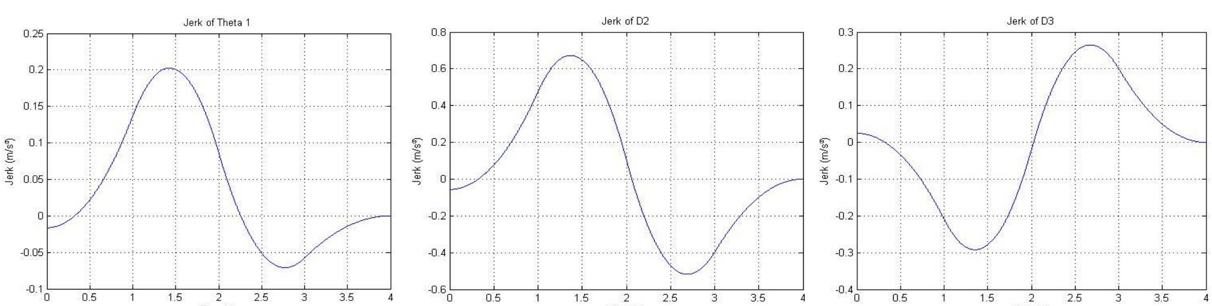


Figure 4. Resulting Jerk within joints

The performance of interpolation points was evaluated considering the maximum deviation between the path and waypoints. Using the parameters $p = 1x10^6$ e h_= 1 s, the maximum measured deviation in joint space was 0,55 µm for the path obtained by interpolation. The value was significantly lower than the maximum errors obtained in regular pneumatic servo tracking applications, which is about 4 mm (Perondi, 2002). Jerk values obtained in the experiment are shown in Table 1.

Table 1. Jerk limits within joints

Robot joints	minimum Jerk	maximum Jerk
Joint 1 (Theta 1)	-0,071 [rad/s^3]	0,2026 [rad/s^3]
Joint 2 (D2)	-0,518 [m/s^3]	0,673 [m/s^3]
Joint 3 (D3)	-0,292[m/s^3]	0,265 [m/s^3]

5. CONCLUSIONS

The method here presented has proved to be useful for environments with obstacles. However, it is observed that, depending on the boundary conditions related to optimization problem, results may not be satisfactory. Concerning the robot presented in this study, the results are not necessarily the best solutions for optimization problem since aspects such as system dynamics and its intrinsic physical features were not considered; the course of action features, Speed and acceleration, for example.

Regarding the traveled distance throughout the path, it can be seen that the obtained path was not necessarily the shortest distance between the start and end point, as this optimization problem does not consider minimizing the distance between the start and end point.

The approximation of points within joint area by using spline to minimize the jerk was considered useful for the achievement of paths which are continuous in speed, acceleration and jerk. In addition, the control parameters of spline, proposed by Simon (2004), made it possible to adjust the path accordingly to task requirements, providing functions with reduced jerk values.

6. REFERENCES

Allgayer, R. S., 2011. "Desenvolvimento de um Manipulador Robótico Cilíndrico Acionado Pneumaticamente". Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Mecânica, UFRGS,

Baba, N and Kubota, N., 1994. "Collision avoidance planning of a robot manipulator by using genetic algorithm. A consideration for the problem in which moving obstacles and/or several robots are included in the workspace." IEEE World Congress on Computational Intelligence., p 714 – 719.

Chyan, G.S and Ponnambalam, S.G.; 2012 "Obstacle avoidance control of redundant Robots using variants of particle swarm optimization" Robotics and Computer-integrated Manufacturing., p. 147-153.

Chong, J.W.S, Ong, S.K, Nee, A.Y.C.N and Youcef-youmi., K, 2009. "Robot Programming using augmented reality: An interactive method for planning collision-free paths." Robotics and Computer-Integrated Manufacturing, p 689 – 701.

Lin, C.-S., Chang, P.R. and Luh, J.Y.S.,1983. " Formulation and optimization of cubic polynomial joint trajectories for industrial manipulators. "IEEE Transctions on Automatic Control. Vol. AC-28., p. 1066-1074.

Missiaggia, L., 2014. "Planejamento Otimizado de Trajetória para um Robô Cilíndrico Acionado Pneumáticamente", Dissertação de Mestrado, UFRGS, 106. p.

Perondi, E. A. Controle Não Linear em Cascata de um Servoposicionador Pneumático com Compensação do Atrito, Tese, UFSC, Florianópolis, 2002.

Rijo, M.G.Q., 2013. "Desenvolvimento da Base e Controle do Grau de Liberdade Rotacional de um Robô Cilíndrico com Acionamento Pneumático", Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Mecânica, UFRGS.

Sarmanho, C.A.C., 2014. "Desenvolvimento de um Robô Pneumático com 5 Graus de Liberdade com Controlador Não Linear com Compensação de Atrito". Tese de Doutorado, UFRGS, 245.p.

Severin, S. Rossmann, J. A, 2012. "Comparison of Different Metaheuristic Algorithms for Optimizing Blended PTP Movements for Industrial Robots." Intelligent Robotics and Applications. p. 321-330.

Slotine, 1988, Putting Physics in Control - The Example of Robotics, IEEE Conttrol System Magazine, Dec 1988, pp.12-17.

Simon, D., 2004. "Data Smoothing and Interpolation Using Eighth-order Algebraic Splines" IEEE Transactions on Signal Processing, Vol. 52.

Tangpattanakul, P., Artrit, P., 2009. "Minimum-time trajectory of robot manipulator using Harmony Search Algorithm". In: 6th International Conference on Electrical Engineering /Electronics, Computer, Telecommunications and Information Technology, pp. 354–357

Tian, L., Collin, C., 2004. "An effective robot trajectory planning method using a genetic algorithm." Original Research Article Mechatronics, Vol.14, p.455-470.

Yang, X.-S., 2010. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2 nd edition.

7. RESPONSIBILITY NOTICE