p-ISSN : 2335-1357 e-ISSN : 2437-1122

Mediterranean Journal of Modeling and Simulation

Med. J. Model. Simul. 06 (2016) 034-044

On-line trajectory planning of time-jerk optimal for robotic arms

Nadir Bendali [a] *, Mohammed Ouali [a]

[a] Structural Mechanics Research Laboratory, Department of Mechanical Engineering, University Blida, Algeria

ARTICLE INFO

Article history :
Received May 2016
Accepted July 2016

Keywords :
Arm vibration ;
Trajectory planning ;
Time intervals ;
Cubic splines ;
Genetic algorithms.

ABSTRACT

A method based on the computation of the time intervals of the knots for time-jerk optimal planning under kinematic constraints of robot manipulators in predefined operations is described in this paper. In order to ensure that the resulting trajectory is smooth enough, a cost function containing a term proportional to the integral of the squared jerk (defined as the derivative of the acceleration) along the trajectory is considered. Moreover, a second term, proportional to the total execution time, is added to the expression of the cost function. A Cubic Spline functions are then used to compose overall trajectory. This method can meet the requirements of a short execution time and low arm vibration of the manipulator and the simulation provides good results.

1. Introduction

The resolve of the problem of time-jerk optimal trajectory planning for robot manipulators along specified tasks is very important, these tasks can be the case of the handling of objects, the drilling/spot welding tasks or the installation of the electronic components. Decreased the execution time of the task is important to increase the productivity of the robot manipulators. Also, limiting the jerk is very important, because high jerk values can wear out the robot structure, and heavily excite its resonance frequencies ; vibrations induced by non-smooth trajectories can damage the robot actuators, and introduce large errors while the robot is performing tasks such as trajectory tracking moreover low-jerk trajectories can be executed more rapidly and accurately.

Many work in the field of robot manipulators has been devoted to study the problem of motion planning along specified tasks, we cite in this context the work of [1-2] the authors have treated the problem of trajectory planning of robot manipulator in imposed tasks by considering the kinematic constraints, they were used the sequential quadratic

*Email :b1102@yahoo.com

programming (SQP) to minimize the cost function which represents a weighting between the execution time of the task and the interval squared jerk, the upper bounds on the absolute values of velocity, be specified, but the total execution time was not set a priori. In [3] a new approach called interval analysis is used to develop an algorithm that minimizes the maximum absolute value of jerk along the trajectory, the cubic splines were used to represent the trajectory in imposed tasks. In [4-5] the authors proposed a fast and unified approach based on particle swarm optimization (PSO) with K-means clustering to solve the near optimal solution of a minimum-jerk joint trajectory constrained by a fixed traverse time of a robot manipulator, the cubic splines were used to interpolate between the nodes of the trajectory in an imposed tasks. In [6-7] the authors has been described an experimental validation of the minimum time-jerk trajectory planning algorithm, the trajectories have been implemented on Cartesian 3-axes manipulator equipped with a piezoelectric accelerometer, the obtained experimental results have been discussed by considering the measure of the acceleration (directly related to the vibration induced on the mechanism) as the comparison parameter. In [8] the authors developed an approach based on fuzzy genetic algorithm using real coding and elitism approach for treat the problem of trajectory planning of robotics arm along specified tasks to minimize time-jerk by considering the kinematic constraints. The authors [9] studied the relationship between the maximum vibratory amplitude and the jerk limit; they formulated the influence of a jerk-controlled movement. In [10] the authors used trigonometric splines to interpolate the trajectories of 3-axes manipulator, where the spline parameters were considered to minimize the jerk by a close form solution.

In this study thus, we propose a unified and fast method for find the minimum time-jerk trajectory planning of robotics arm based on the computation of the time intervals of the knots. To validate the proposed method, an objective function [8] is used and the results demonstrate that our proposed method achieves the best results by less computation time than [8].

This paper is organized as follows. In Section 2, the minimum time-jerk joint trajectory optimization problem is formulated. In section 3, the kinematic constraints are presented. In section 4, the proposed method based on the time intervals and genetic algorithms is discussed. In section 5, numerical application on a three degree-of-freedom glass substrate handling robot is presented. Discussions and comparisons results are presented in Section 6. Conclusions are summarized in Section 7.

2. Problem formulation

In our joint trajectory planning, we assume the robot manipulator has N joints and the trajectory for each joint has M+1 knot points including the first and last. Thus, for each joint, there exist M time intervals and we choose the second and penultimate knot points as the extra points to represent the robot trajectory [11].

We apply cubic splines function to each joint to interpolate the joint trajectory between every two neighbor knot points. The velocities and accelerations of the initial and terminal conditions (v_1, v_m, a_1,and a_m) are specified to be zero. These conditions cause two equations of the spline algorithm becoming zero and the path pattern cannot be solved. Therefore, two extra knots (position values at time t_2 and t_{m-1}) are added and their position values are not specified. Let $Q_i(t_i)$ be the cubic polynomial for the $i-th$ joint in the time intervals $[t_j, t_{j+1}]$. The second derivative of $Q_i(t_i)$ is a linear interpolation and can be written as [4] :

$$\ddot{Q}_i(t) = \frac{t_{i+1}-t}{h_i}\ddot{Q}_i(t_i) + \frac{t-t_i}{h_i}\ddot{Q}_i(t_{i+1}); \qquad i = 1, 2, ..., m-1 \tag{1}$$

Where, $h_i = t_{i+1} + t_i$

Integrating equation (1) for the given points $Q_j(t_i) = q_i$ and $Q_i(t_{i+1}) = q_{i+1}$, the following interpolation functions are obtained :

$$\dot{Q}_i(t) = -\frac{\ddot{Q}_i(t_i)}{2h_i}(t_{i+1}-t)^2 + \frac{\ddot{Q}_j(t_{i+1})}{2h_i}(t-t_i)^2 + \left[\frac{q_{i+1}}{h_i} - \frac{h_i\ddot{Q}_i(t_{i+1})}{6}\right] - \left[\frac{q_i}{h_i} - \frac{h_i\ddot{Q}_i(t_i)}{6}\right] \tag{2}$$

And

$$Q_i(t) = \frac{\ddot{Q}_i(t_i)}{6h_i}(t_{i+1}-t)^3 + \frac{\ddot{Q}_i(t_{i+1})}{6h_i}(t-t_i)^6 + \left[\frac{q_{i+1}}{h_i} - \frac{h_i\ddot{Q}_i(t_{i+1})}{6}\right](t-t_i) + \left[\frac{q_i}{h_i} - \frac{h_i\ddot{Q}_i(t_i)}{6}\right](t_{i+1}-t) \tag{3}$$

Then, the two extra knots positions values q_2 and q_{m-1} are not fixed and are used to add two new equations to the system in such a way that it can be solved. The joint displacements of these two knots are written as :

$$q_2 = q_1 + h_1 v_1 + \frac{h_1^2}{3}a_1 + \frac{h_1^2}{6}\ddot{Q}_2(t_2) \tag{4}$$

$$q_{m-1} = q_m - h_{m-1}v_m + \frac{h_{m-1}^2}{3}a_m + \frac{h_{m-1}^2}{6}\ddot{Q}_{m-1}(t_{m-1}) \tag{5}$$

Using the continuity conditions on velocities and accelerations, a system of $m-2$ linear equations solving for $m-2$ unknowns $\ddot{Q}_i(t_i)$, $i = 2, 3, ..., m-1$ is obtained as :

$$A\left[\ddot{Q}_2(t_2) \quad \ddot{Q}_3(t_3) \ldots \ddot{Q}_{m-1}(t_{m-1})\right]^T = B \tag{6}$$

In (6), the matrix A is non-singular matrix and entries of the vector B are changed for each joint.

Where,

$$A = \begin{bmatrix} 3h_1 + 2h_2 + h_1^2/h_2 & h_2 & & & & \\ h_2 - h_1^2/h_2 & 2(h_2+h_3) & h_3 & & \emptyset & \\ & h_3 & 2(h_3+h_4) & h_4 & & \\ & & & \vdots & & \\ & \emptyset & & h_{m-3} & 2(h_{m-2}+h_{m-3}) & K_1 \\ & & & & h_{m-2} & K_2 \end{bmatrix}$$

Where,

$$K_1 = \left(h_{m-2} - \frac{h_{m-1}^2}{h_{m-2}} \right), \; K_2 = \left(3h_{m-1} + 2h_{m-2} + \frac{h_{m-1}^2}{h_{m-2}} \right)$$

And the vector B is given by :

$$B = \begin{bmatrix} 6(q_3/h_2 + q_1/h_1) - 6(1/h_1 + 1/h_2)\left[q_1 + h_1\dot{q}_1 + (h_1^2/3)\ddot{q}_1\right] - h_1\ddot{q}_1 \\ (6/h_2)\left[q_1 + h_1\dot{q}_1 + (h_1^2/3)\ddot{q}_1\right] + 6q_4/h_3 - 6(1/h_2 + 1/h_3)q_3 \\ 6\left[(q_{i+1} - q_i)/h_i - (q_i - q_{i-1})/h_{i-1}\right] \\ \vdots \\ (6/h_{m-2})\left[q_m - h_{m-1}\dot{q}_m + (h_{m-1}^2/3)\ddot{q}_m\right] \\ -6(1/h_{m-2} + 1/h_{m-3})q_{m-2} + 6q_{m-3}/h_{m-3} \\ -6(1/h_{m-1} + 1/h_{m-2})\left[q_m - h_{m-1}\dot{q}_m + (h_{m-1}^2/3)\ddot{q}_m\right] \\ +6(q_m/h_{m-1} + q_{m-2}/h_{m-2}) - h_{m-1}\ddot{q}_m \end{bmatrix}$$

After these $M+1$ parameters are solved, we derive the jerk of the trajectory of the $i-th$ joint by the equation :

$$J_i(t) = -\frac{1}{h_i}\ddot{Q}_i(t_i) + \frac{1}{h_i}\dddot{Q}_i(t_{i+1}); \qquad i = 1, 2, ..., m-1 \tag{7}$$

From (7), we determine that the jerk depends on the length of the time interval h_i. Once we derive the jerk, the minimum jerk joint trajectory optimization problem can be formulated by an objective function described as : solve the maximum value of the jerk of each joint along the trajectory and minimize the summation of every maximum value of the jerk that is :

$$\min \left[\sum_{i=1}^{N} \max_{h_i} |J_i(t)| \right] \tag{8}$$

The jerk is the key factor that causes robot arm vibration mainly when decreased the execution time of the task is important. In the paper, we adopt the objective function used in [8] which minimize two terms composed by the term proportional to the total execution time and the other one proportional to the total jerk. So the mathematic expression of the objective function model can be defined as :

$$F_{obj} = \min \left(k_T \sum_{i=1}^{N} T_i + k_J \sum_{i=1}^{N} \max_{h_i} |J_i(t)| \right) \tag{9}$$

The subjects are :

– Joint velocities : $\left|\dot{Q}_{ij}(t)\right| \leq \dot{Q}_{ij}^{\max}$ for $i = 1, ...n$ and $j = 1, ...m-1$

– Joint accelerations : $\left|\ddot{Q}_{ij}(t)\right| \leq \ddot{Q}_{ij}^{\max}$ for $i = 1, ...n$ and $j = 1, ...m-1$

– Joint jerks : $|J_{ij}(t)| \leq J_{ij}^{\max}$ for $i = 1, ...n$ and $j = 1, ...m-1$

Here, T_i is the total execution time of the tasks ; k_T and k_J are the weight coefficient change according to the user needs can favor either the execution time of the task is the jerk. Also, $\dot{Q}_{ij}^{\max}$, $\ddot{Q}_{ij}^{\max}$ and $J_{ij}^{\max}$ are limit kinematics performances of the $i-th$ joint deduced from technological and design data.

3. Constraints formulation

The velocity constraints of the optimization problem are formulated into the maximum absolute value of velocities at the extreme points t_i and t_{i+1} or $\overset{\otimes}{t}_i$ where $\overset{\otimes}{\dot{Q}}_{ji} = \dot{Q}_{ji}\left(\overset{\otimes}{t}_i\right) = 0$ in each interval [2]. The velocity is calculated using equation (2). The velocity constraints become :

$$\max\left\{\left|\dot{Q}_{ji}\left(t_i\right)\right|, \left|\dot{Q}_{ji}\left(t_{i+1}\right)\right|, \left|\overset{\otimes}{\dot{Q}}_{ji}\right|\right\} \leq \dot{Q}_j^{\max}; \qquad j = 1, ..., N; \quad and \quad i = 1, ..., m-1 \quad (10)$$

The acceleration is the solution of system (6). The acceleration constraints are formulated from the acceleration linear function and the maximum absolute value exists at t_i or t_{i+1}. The acceleration constraints become :

$$\max\left\{\left|\ddot{Q}_{j,1}\right|, ..., \left|\ddot{Q}_{j,n}\right|\right\} \leq \ddot{Q}_j^{\max}; \qquad j = 1, ..., N \qquad (11)$$

4. Proposed approach based in Genetic Algorithm

4.1. Initialization and route of generation

Let h be defined as the vector of design variables $h_i = [h_1, h_2, ..., h_{m-1}]$. To initialize the optimization process it is considered that :

$$h^{(0)} = \max_{j=1,...,m-1}\left[\left(\frac{|q_{i2} - q_{i1}|}{\dot{Q}_i^{\max}}\right), \left(\frac{\left|\dot{Q}_{i2} - \dot{Q}_{i1}\right|}{\ddot{Q}_i^{\max}}\right), ..., \left(\frac{\left|\ddot{Q}_{im} - \ddot{Q}_{i,m-1}\right|}{J_i^{\max}}\right)\right]$$

As two extra knots are needed they are initially taken as :

$q_{i2} = (q_{i1} + q_{i3})/2$ and $q_{i,m-1} = (q_{j,m-2} + q_{im})/2$

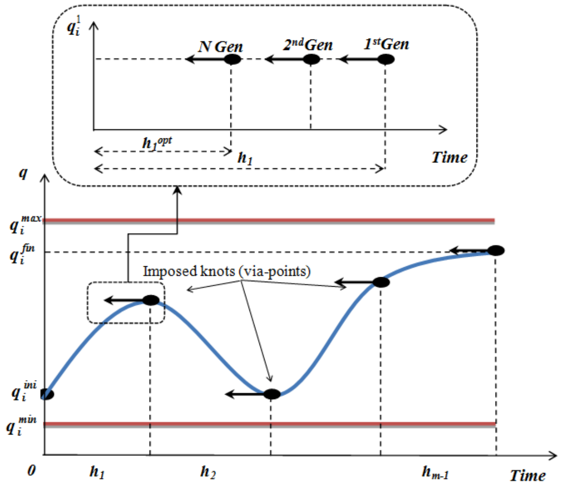


Fig. 1 – Representation of the Cubic Spline Trajectory with the horizontal movement of the intermediate knots and the end point after generation.

During the optimization process the intermediate knots and the end point will generate only horizontally as seen in Fig.1, consequently the trajectory changes and moves also horizontally by minimizing the objective function and obtaining the best optimal vector as :

$$h^{opt} = \left[h_1^{opt}, h_2^{opt}, \ldots, h_{m-1}^{opt}\right] \tag{12}$$

4.2. Genetic algorithms

The use of a genetic algorithm starts with the creation of an initial population or chromosome in genetics, this chromosome is composed by genes or their number (G) is defined according to the number of the knots (k) used to generate the trajectories where $k = G + 1$; it should be noted that this stage requires a coding of the genes, for that we have a real coding of these genes.

Thereafter this initial population will be generated by chance and we obtain for each chromosome a solutions corresponding with his performance index. For create the next generation, three genetic operators are applied :

- Reproduction : Usually the general strategy of reproduction is that the chromosomes (parents) with better performance index have the possibility of reproducing more.
- The crossover. It is the operator who will allow the mixing of the genetic characters of the population, this operator will create two children by carrying out a mixture of the chromosomes of two parents. In the simulation we will fix the rate or the probability of crossover equal to 65%.
- The mutation. It consists in deteriorating the coding of a chromosome. Its role is to make emerge new genes by exploring zones of the space of research which could not be visited by simple application of the operator of crossover ; in practice there exist many manners of transferring a chromosome by the modification of one or more gene, or by change of position of a gene, or the suppression by adding a gene. In the simulation we will fix the probability of mutation equal 4%.

5. Numerical application

The numerical application was implemented on a three degree-of-freedom glass substrate handling robot considered by authors in [8] to find the near optimal solution of a minimum-jerk joint trajectory using our proposed method.

The trajectory was given by four knot points and five time intervals ($M = 5$) for all the three joints. Table 1 shows the interpolation point positions of each joint and Table 2 shows the maximum kinematic limits of velocity and acceleration of each joint.

Table 1 – Interpolation point of each joint.

	Interpolation point (deg)					
Joint	1	2	3	4	5	6
1	120	Virtual	90	45	Virtual	0
2	-10	Virtual	60	40	Virtual	100
3	0	Virtual	-20	30	Virtual	70

Table 2 – Kinematics constraints of each joint.

Joint	Velocity (deg/s)	Acceleration (deg/s^2)
1	100	70
2	95	75
3	100	75

To represent the trajectory and solve the two unknown extra knot points in Table 1 we use both Eq. (4-5). We computed the jerk J_{ii} of the trajectory by Eq. (7) and formulated the minimum-jerk optimization problem by Eq. (9). In this optimization problem, the solution was transformed and denoted as Eq. (14). In the beginning, we created the initial population (chromosomes), and to seek the optimal trajectory, we must generate by chance according to the genetic algorithms technique a trajectories, the latter candidate of chromosome will be evaluated then compared with others, this operation is repeated for all the introduced chromosomes, and the best result is which satisfies the given objective function. It should be noted that any trajectory which would violate one of the velocities or accelerations constraints indicated in Table 2 will be automatically rejected by Eq. (10-11) respectively. These results were validated by the results obtained in [8].

6. Comparison of results and discuss

Each joint of cubic spline trajectories including their derivatives (velocities, accelerations and jerk) for $K_J/K_T = 0.02$ and $K_J/K_T = 0.3$ are illustrated in Fig. 2(a) and Fig. 2(b) respectively. For the first case we obtained a vector of time intervals $h_i = [0.6017, 1.6992, 1.4349, 1.2623, 1.0895]$ equal to execution time $\sum_{i=1}^{5} h_i = 6.0876\,\text{sec}$, for the second case we obtained a vector of time intervals $h_i = [1.0301, 2.9256, 2.3687, 1.9771, 1.4761]$

equal to execution time $\sum_{i=1}^{5} h_i = 9.777\,\text{sec}$ this results show that the technique described in this paper obtains the better solution than [8] used Fuzzy genetic algorithm which obtained the execution time of task equal $\sum_{i=1}^{5} h_i = 6.72\,\text{sec}$ for the first case and $\sum_{i=1}^{5} h_i = 10.262\,\text{sec}$ for the second case.

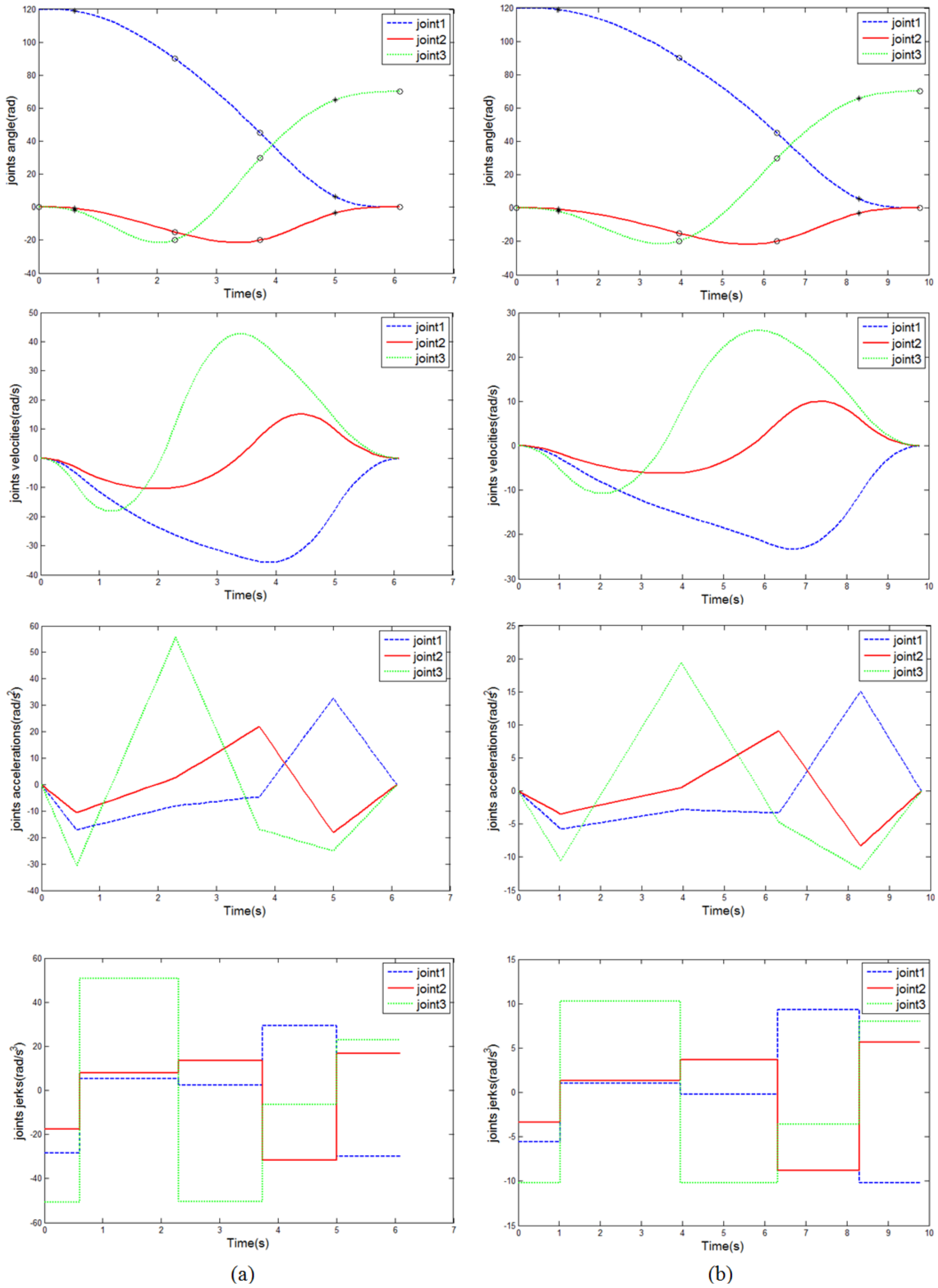
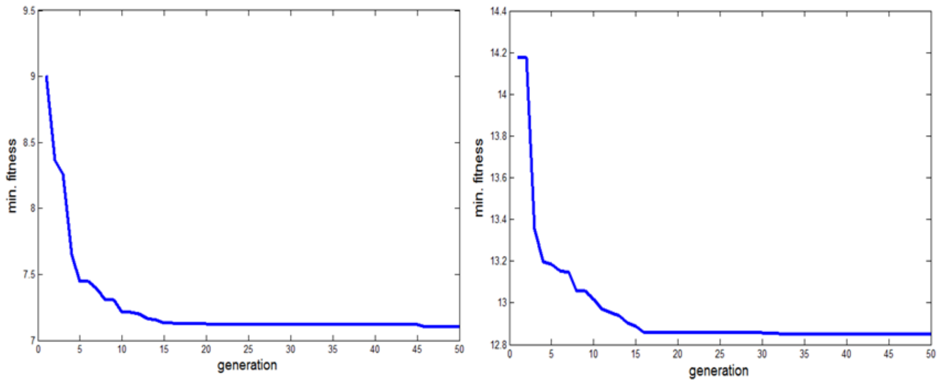


Fig. 2 – The results of the simulation showing the trajectories of each joints angles (the circles indicate the knot positions and the crosses indicate the two dummy knots), velocities, accelerations and jerks of the glass substrate handling robot (a) : $K_J/\ K_T = 0.02$ and (b) : $K_J/\ K_T = 0.03$

Fig.3 shows the results histories of the optimization process of glass substrate handling robot for $K_J/K_T = 0.02$ and $K_J/K_T = 0.3$, for the first case we obtained an objective function $F_{obj} = 7.1025$ and for the second case the objective function $F_{obj} = 12.846$. These results were obtained with the fixed number of generation is 50 corresponding of 1/10 number of generation used in [8] ; meanwhile our proposed method converges quickly and have faster calculation speed.



(a) (b)

Fig. 3 – Result histories of the proposed method of the glass substrate handling robot (a) : $K_J/\ K_T = 0.02$ and (b) : $K_J/\ K_T = 0.03$

Table 3 – Results of the optimal time intervals and the maximum jerks obtained by our proposed method and compared with [8].

K_J/K_T	Proposed method				Cong's method [8]			
	h_i (s)	$J_1^{\max}$ $(°/s^3)$	$J_2^{\max}$ $(°/s^3)$	$J_3^{\max}$ $(°/s^3)$	h_i (s)	$J_1^{\max}$ $(°/s^3)$	$J_2^{\max}$ $(°/s^3)$	$J_3^{\max}$ $(°/s^3)$
0.5	11.483	6.33	5.56	6.32	11.484	5.235	6.467	11.60
0.4	10.284	8.71	7.85	8.74	10.976	5.764	7.152	13.83
0.3	9.777	10.22	8.82	10.23	10.262	6.714	8.399	17.85
0.2	8.665	14.78	12.75	14.80	9.371	8.088	10.21	25.94
0.15	8.389	15.99	14.63	16.11	8.939	9.486	11.97	30.31
0.1	7.433	23.86	18.21	23.86	8.111	12.60	15.97	43.40
0.05	6.884	29.66	23.15	30.08	7.13	17.78	25.46	70.65
0.02	6.087	29.94	31.77	50.74	6.72	27.72	46.01	79.28
0	4.896	79.76	58.89	99.86	6.29	99.21	135.6	103.0

Table 3 reports the maximum jerk values resulting from the optimization procedure with different values of weighting coefficients K_J/K_T ; such values are compared with those yielded by the method proposed by [8]. Considering the value of K_J/K_T if it is over 0.3 the robot will be put more time for execution of the task and if the value is less of

0.02 the robot will undergo more of vibration in their joints; so the optimal values are between the interval [0.02, 0.3]. It can be noticed that the results yielded by the method described in this paper are well comparable with those provided by the method [8] with respect to the maximum values of velocity and acceleration.

7. Conclusion

This research work deals with optimal trajectory planning using time intervals method to solve the problems of optimal imposed motion of robotics arm.

Minimum objective function computed and it is composed of two terms : the first term is proportional to the total execution time directly affected production efficiency and the latter term is proportional to the maximum jerk of each joint based on the minimax approach that ensures the optimal trajectory is smooth enough, taking into account the main constraints imposed on the robot kinematic (velocity and acceleration) performance. The trajectories were modeled using Cubic splines functions who allow guaranteeing the smoothing of the trajectory and at the same time the continuity of velocities, accelerations and the jerks.

Finally, the proposed method has been run in simulation, taking as input data those found in the work by [8]. Comparison of the results with those provided in [8] has shown that the effectiveness of our method is effective in performing an optimal trajectory planning to solve the problem between high production efficiency and low structure vibration.

This work opens the door for further investigations such as using the B-Spline functions or Non Uniform Rational B-Spline (NURBS) functions and considering an obstacle in workspace and taking account the dynamic constraints of the industrial robots, so as to evaluate the applicability the proposed method and its results.

REFERENCES

[1] Gasparetto, A., Zanatto, V., 2007. A New Method for Smooth Trajectory Planning of Robot Manipulators, Mechanism and Machine Theory 42, p455-471.

[2] Gasparetto, A., Zanatto, V., 2008. A technique for time-jerk optimal planning of robot trajectories, Robotics and Computer-Integrated Manufacturing 24, 415–426.

[3] Piazzi, A., Visioli, A., 2000. Global minimum-jerk trajectory planning of robot manipulators, IEEE Transactions on Industrial Electronics 47 (1), 140-149.

[4] Lin, H., 2014. A Fast and Unified Method to Find a Minimum-Jerk Robot Joint Trajectory Using Particle Swarm Optimization, Journal of Intelligent & Robotic Systems - Springer.

[5] Lin, H., Liu, Y. C., 2011. “Minimum-Jerk Robot Joint Trajectory Using Particle Swarm Optimization,” First International Conference on Robot, Vision and Signal Processing.

[6] Zanotto, V., Gasparetto, A., Lanzutti, A., Boscariol, P., Vidoni, R.,2011. Experimental Validation of Minimum Time-jerk Algorithms for Industrial Robots, J Intell Robot Syst, 64, 197–219.

[7] Gasparetto, A., Lanzutti, A., Vidoni, R., Zanotto, V., 2012. Experimental Validation and Comparative Analysis of Time-jerk Algorithms for Trajectory Planning, Robotics and Computer-Integrated Manufacturing 28, 164-181.

[8] Cong, M., Xu, X., Xu, P., 2010. Time-jerk synthetic optimal trajectory planning of robot based on fuzzy genetic algorithm, Int. J. Intelligent Systems Technologies and Applications, Vol. 8, Nos. 1-4.
[9] Barre, P, J., Bearee, R., Borne, P., Dumetz, E., 2005. Influence of a jerk controlled movement law on the vibratory behavior of high-dynamics systems. J. Intell. Robot. Syst. 42, 275–293.
[10] Simon, D., Isik, C., 1993. A trigonometric trajectory generator for robotic arms. Int. J. Control 57(3), 505–517.
[11] Lin, C, S., Chang, P, R., Luh, J, Y, S., 1983. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. IEEE Trans. Autom. Control 28(12), 1066–1073.