ANNALS OF FACULTY ENGINEERING HUNEDOARA
- INTERNATIONAL JOURNAL OF ENGINEERING
Tome IX (Year 2011). Fascicule 1. (ISSN 1584 - 2665)

[1.] Albano LANZUTTI

SMOOTH TRAJECTORY PLANNING ALGORITHMS FOR INDUSTRIAL ROBOTS: AN EXPERIMENTAL EVALUATION

[1.] DIPARTIMENTO DI INGEGNERIA ELETTRICA, GESTIONALE E MECCANICA
UNIVERSITA' DI UDINE, UDINE – ITALY

ABSTRACT: An analysis of the experimental results of a new method for smooth trajectory planning for robot manipulators is presented in this paper. The technique is based on the minimization of an objective function that is composed of two terms: the first one is proportional to the trajectory execution time, the second one is proportional to the integral of the squared jerk. The need for a smooth trajectory and the need for a fast execution can be adjusted by changing the values of two constants that weigh the two terms. The trajectory execution time is not set *a priori* and the kinematic constraints on the robot motion are taken into account. Cubic splines and fifth-order B-splines are used to compose the overall trajectory. Two different trajectory planning techniques (the first one minimizes the maximum absolute value of the jerk along the whole trajectory, while the second one ensures only the continuity of the position, velocity and acceleration values) have been implemented with the aim to compare the outcomes of the tests. The described methods are applied to a 3-d.o.f. Cartesian robot and the experimental tests are carried out by using an accelerometer mounted on the manipulator end-effector.
KEYWORDS: Trajectory planning, Smoothness, Time-jerk optimization, Experimental validation

❖ INTRODUCTION

One of the most important current robotic industrial requirements is the estimation and the reduction of the manipulators vibrational phenomena. Indeed, the demand for increasing productivity through fast and high precision motion is growing, thus the designers are forced to reduce the masses of the robot structures, resulting in a loss of structural rigidity and an increase of flexibility affecting also the dynamic response of the system.

A proper calibration of the manipulator control system and a dedicated action on the trajectory planning phase [1] can be considered as a solution of the problem.

The trajectory planning is a fundamental issue for robotics applications and automation in general. At high operating speeds, required in many current tasks, the possibility to generate trajectories that satisfy specific targets and requirements is a basic step to ensure optimal results. Robotic movements and trajectories that have smoothness properties are becoming more widely used in modern applications. Indeed, the planning of trajectories with a bounded value of the jerk is an important target, since this allows to reach higher task execution speeds, reduce the excitation of the resonant frequencies of the manipulator structure and improve the tracking accuracy.

The analysis of the scientific literature shows that the trajectory planning problem is based on the optimization of some objective function or of some parameters. Criteria that are based on minimum execution time, minimum energy or actuator effort, minimum jerk or hybrid optimality approaches can be found.

With the aim to increase the productivity in the industrial sector, the first trajectory planning techniques proposed were the minimum-time algorithms. Starting from unconstrained problems [2], this type of optimization is recently evolved in minimum time algorithms under kinematic constraints (i.e. maximum values for velocity, acceleration and jerk) [3].

A second important criterion for trajectory planning is focused on the minimization of the actuator effort, i.e. the minimization of the energy required to the manipulator actuators [4]. If the energy consumption is minimized instead of the execution time, the effort of the actuators and the stresses of the manipulator are reduced, moreover the resulting trajectory is easier to track. This type of optimization is then preferable in applications with limited capacity of energy source.

Another type of trajectory planning algorithms is based on the optimization of the jerk along the whole length of the path [5-6]. If this technique is used, the excitation of the resonant frequencies of the mechanical system is reduced. Thus, the stresses to the actuators and to the robot structure are intrinsically reduced, and the tracking errors decrease.

As mentioned in the foregoing, starting from the fundamental optimization techniques above described, hybrid optimality approaches are implemented.

For instance, hybrid time-energy-optimal trajectory planning algorithms can be found in [7]. With the aim to reach the advantages of the jerk reduction in fast trajectories, hybrid time-jerk optimal techniques are proposed [8-12]. These algorithms are based on different primitives that are used to interpolate the path (e.g. trigonometric splines in [8], polynomials of fourth and fifth order in [10]) and different optimization procedures (e.g. genetic algorithms are used in [9], SQP algorithm in [11-12].

One of the most popular algorithms for planning smooth trajectories is described in [5-6]. Based on interval analysis, this technique minimizes the absolute maximum value of the jerk along a trajectory whose execution time is known and set *a priori*. Cubic splines are used to interpolate the via-points of the path and the output of the algorithm is a set of time intervals that produces the lowest jerk peak. A minimum time-jerk trajectory planning technique is presented in [11-12]. Two algorithms based upon a minimization of an objective function that takes into account the speed and the smooth of the trajectory are presented. More in detail, the objective function is composed of a term that is proportional to the total execution time and of a term that is proportional to the integral of the squared jerk along the path, both weighted by two parameters. A method based on the objective function defined in [11-12] and extended by considering also the power consumption of the actuating motors and the joints physical limits (so that the technique is a time-jerk-energy planning algorithm) is presented in [13].

In this paper, the two trajectory planning algorithms presented in [11-12] are considered. Unlike most jerk-minimization techniques, this method does not require a trajectory execution time set *a priori*, and takes into account the robot motion constraints. Thus, one can define the upper bounds on the absolute values of velocity, acceleration and jerk for each robot joint. In order to demonstrate the benefits of the used algorithms (i.e. reduced mechanical stresses and reduced vibrational phenomena), the trajectories so planned are input to a 3-d.o.f. Cartesian robot and the vibrations of its arms during their movements are evaluated by using an accelerometer.

With the aim of evaluating the results obtained with the minimum time-jerk technique, both the method described in [5-6] and a classical spline based planning algorithm have been implemented and experimentally tested on the Cartesian manipulator.

The paper is organized as follows: in section 2 the planning algorithms [11-12] and [5-6] and the main characteristics of the planning techniques under test are explained; the simulations and experimental results of the used techniques, with a brief description of the experimental set-up, are analyzed in section 3.

❖ THE TRAJECTORY PLANNING ALGORITHMS. MINIMUM TIME-JERK TRAJECTORY PLANNING ALGORITHM

The minimum time-jerk algorithm (described with many details in [11-12]) concerns trajectories *off-line* geometrically defined. Accordingly, a *path planner* at the top level generates the geometric paths (obstacle avoidance problems are solved at this level) as a sequence of nodes in the operative space which represent successive positions and orientations of the end-effector of the manipulator. The execution time of the planned trajectory is not set *a priori (*it is a result of the algorithm), and the upper bounds on velocity, acceleration and jerk (the kinematic constraints) are taken into account.

The generated trajectory is optimized in the sense of the best compromise between execution time and value of the jerk. In order to achieve this task, a hybrid objective function made of two terms having opposite effects is considered. The first term is proportional to the trajectory execution time, whereas the second term is proportional to the integral of the squared jerk. The two effects are weighted by the coefficients k_T and k_J respectively.

In order to represent the trajectories, two specific primitives are chosen. The first primitive is a cubic spline (the algorithm is so called SPL3J) and the objective function is given by:

$$FOBJ = k_T \sum_{i=1}^{n-1} h_i + k_J \sum_{j=1}^{N} \sum_{i=1}^{n-1} \left[\frac{\left(\alpha_{j,i+1} - \alpha_{j,i}\right)^2}{h_i} \right] \quad (1)$$

where $\alpha_{j,i}$ is the acceleration of the j-th joint at the i-th via-point, n is the number of the via-points of the path, N is the number of robot joints and h_i is the time interval between two via-points (for more details [12]).

The second primitive is a fifth-order B-spline, degree p = 5 and order k = 2r = 6, (the algorithm is so called BSPL5J) and the objective function is given by:

$$FOBJ = k_T \sum_{i=1}^{vp+1} h_i + k_J \sum_{j=1}^{N} \int_0^{tf} \left(\sum_{k=1}^{n-2} CPJ_{j,k} \cdot N_{p-3,k}(t) \right)^2 dt \quad (2)$$

where vp is the number of via-points, n+1 is the number of control points (n = vp + 2(r-1)), $N_{i,p}(t)$ is the base function, $CPJ_{j,k}$ is the control point of the jerk and t_f is the total execution time of the trajectory (for more details [11]).

The solution of the optimization problem is a vector of time intervals h_i between two subsequent via-points that minimize the objective functions (1) or (2).

With a suitable choosing of the value of the two weights k_T and k_J, in both solver methods above described, a balance between speed and smoothness of the trajectory can be reached. The limit conditions are the minimum execution time (i.e. $k_J = 0$) and the minimum jerk value (i.e. $k_T = 0$). A criterion to make the choice of the two weights is reported in [14].

❖ GLOBAL MINIMUM JERK TRAJECTORY PLANNING ALGORITHM

For a comparative analysis of the experimental results, the global minimum jerk trajectory planning algorithm (so called GMJ) presented in [5-6] has been implemented. In this technique, the execution time of the trajectory is set *a priori* and the manipulator kinematic constraints are not taken into account. Moreover, cubic splines are used to interpolate the sequence of points of the geometric path that is planned in *off-line* mode.

The algorithm can be summarized as follows. If **h** is the vector of the time intervals between two consecutive via-points, and defined $j_{k,i}(\mathbf{h})$ as the value of jerk of the i-th spline at the k-th joint, the optimization problem of the GMJ algorithm is :

$$\min_{h \in \mathbf{R}^{+n}} \max \{ |j_{k,i}(\mathrm{h})| : i = 1, \ldots, n; k = 1, \ldots, m \} \tag{3}$$

subject to:

$$\sum_{i=1}^{n} h_i = T \tag{4}$$

where n is the number of via-points, m is the number of robot joints and T is the trajectory execution time. The output of the algorithm is a set of time intervals h_i that minimizes the absolute maximum value of the jerk along the whole path.

❖ COMPARATIVE ANALYSIS OF THE PLANNING TECHNIQUES PROPERTIES

With the aim of evaluating the trajectories obtained by running the three techniques above described (SPL3J, BSPL5J and GMJ), a fourth algorithm has been implemented. It is based on cubic splines (so as to ensure the continuity of position, velocity and acceleration values). The duration of the time intervals between two via-points is proportional to the trajectory execution time, that is set *a priori*, and the number of via-points (accordingly the algorithm is called PROP). The algorithm so implemented does not take into account the kinematic constraints of the manipulator. In Table I the main properties of the four algorithms are reported. An important remark on the convergence time of the four techniques that have been used must be done: for all the tested trajectories, GMJ algorithm gives the solution after several hours, whereas SPL3J, BSPL5J and PROP algorithms take less than a minute to produce the solutions. This drawback is very important if, for example, the techniques will be used to plan trajectories for industrial applications where short times of solution are necessary.

Table I : Main properties of the SPL3J, BSPL5J, GMJ and PROP algorithms

Algorithm	Primitive	Trajectory time	Optimization	Kinematic Constraints
SPL3J	cubic spline	calculated	jerk-time	Yes
BSPL5J	quintic B-spline	calculated	jerk-time	Yes
GMJ	cubic spline	imposed	max jerk	No
PROP	cubic spline	imposed		No

❖ EXPERIMENTAL EVALUATION OF VIBRATIONAL PHENOMENA. IMPLEMENTED TRAJECTORIES

Three different trajectories have been implemented in Matlab™ and then input to the Cartesian manipulator with the aim to test and validate the benefits of using smooth trajectory planning algorithms. The target of the experimental tests is to compare the vibrational phenomena on the robot end-effector that are induced by the movements of its arms after applying the four techniques above described on the same geometric path. This means that the trajectories via-points and the execution time associated to each path are the same for each algorithm. In order to reach the second target, the three trajectories are first simulated with the SPL3J and BSPL5J (the values of k_T and k_J are set with the aim to get the same execution time) and the execution time so obtained is then input in the GMJ and PROP algorithms. In this way, with the same test starting conditions for the four techniques, a more strict evaluation of the vibrational phenomena can be conducted. The three trajectories are below described:

Trajectory #1: the first path implements a pick-and-place movement. The k_T and k_J values are respectively 860 and 0.005 for SPL3J technique, 10 and 1 for BSPL5J and the execution time is 7.4 s. The via-points of the trajectory are reported in Table II and in Table III the simulated mean and maximum jerk values for each algorithm are included.

Table II : Trajectory #1 via-points

Via-points	X position [mm]	Y position [mm]	Z position [mm]
1	0	0	0
2	0	0	-170
3	10	12.9	-190
4	30	38.6	-200
5	175	225	-200
6	320	411.4	-200
7	340	437.1	-190
8	350	450	-170
9	350	450	0

Table III : Maximum and mean jerk values for the four algorithms

	max [mm/s^3]			mean [mm/s^3]		
	x	y	z	x	y	z
SPL3J	183.54	230.43	322.49	90.04	115.68	182.84
BSPL5J	178.90	231.13	376.55	74.90	96.16	178.77
GMJ	169.52	212.19	305.21	88.90	114.18	193.02
PROP	588.30	756.70	1105.20	264.32	340.06	533.65

Trajectory #2: the second example implements a "L-shaped" path. The k_T and k_J values are respectively 845 and 0.005 for SPL3J technique, 139 and 1 for BSPL5J and the execution time is 5 s. In Table IV the trajectory via-points are reported. For each technique, the simulated maximum and mean jerk values are shown in Table V.

Table IV : Trajectory #2 via-points

Via-points	X position [mm]	Y position [mm]	Z position [mm]
1	0	0	0
2	270	0	20
3	290	0	40
4	290	20	60
5	290	290	80

Table V : Maximum and mean jerk values for the four algorithms

	max [mm/s^3]			mean [mm/s^3]		
	x	y	z	x	y	z
SPL3J	531.75	532.01	195.93	278.86	278.22	84.96
BSPL5J	615.39	696.89	189.58	244.31	215.50	64.04
GMJ	511.99	524.35	184.15	273.33	293.09	56.49
PROP	1259.40	1259.20	34.60	474.02	473.92	23.04

Trajectory #3 : the last trajectory is a square with five via-points, whose sequence is reported in Table VI. The k_T and k_J values are respectively 1280 and 0.5 for SPL3J technique, 10 and 1 for BSPL5J and the execution time is 14.5 s.

Table VI : Trajectory #3 via-points

Via-points	X position [mm]	Y position [mm]	Z position [mm]
1	10	10	0
2	330	10	0
3	330	330	-170
4	10	330	-170
5	10	10	0

The simulated maximum and mean jerk values for the four algorithms are reported in Table VII. As mentioned before, the SPL3J and BSPL5J algorithms optimize the trajectories in the sense of best trade-off between the execution time and the integral of the squared jerk, whereas the GMJ technique minimizes the absolute maximum value of the jerk along the path. Starting from these considerations, the lowest maximum values of the jerk and the lowest mean jerk values are expected if the GMJ and SPL3J/BSPL5J are used respectively.

Table VII : Mean jerk values for the four algorithms

	max [mm/s^3]			mean [mm/s^3]		
	x	y	z	x	y	z
SPL3J	68.96	61.95	32.91	35.39	32.38	17.26
BSPL5J	86.44	69.30	36.82	30.25	27.66	14.70
GMJ	49.42	57.28	30.43	32.35	34.11	18.12
PROP	82.59	82.59	43.88	38.87	38.87	20.65

If Tables III, V and VII are considered, it is possible to find a confirmation to the above anticipations: the GMJ algorithm provides the lowest maximum jerk values if compared with the other three techniques, while SPL3J and BSPL5J feature the lowest mean jerk values. Then, it is possible to verify that the PROP method is the worst in terms of both the mean and the maximum jerk values.

❖ EXPERIMENTAL SET-UP

The experimental tests, aimed to evaluate the vibration phenomena during the execution of the three trajectories planned with the four algorithms, are made on a Cartesian manipulator (Figure 1), controlled using a real time external controller.

The 3-d.o.f. manipulator has three prismatic joints, whose kinematic bounds are shown in Table VIII, a workspace of 500x600x500 mm (X, Y and Z) and an accuracy of 0.1 mm. The joints are actuated by means of brushless servo-motors, coupled with the robot arms by using a cogged belt and equipped with resolver position sensors. Each motor is linked to the transmission belt by a reduction gear head.

An embedded multifunction board, the Sensoray S626, is used in order to realize a link between the manipulator and the external control loop. The position real time controller is set up on an AMD Athlon(tm) XP 2400 (1.99 GHz with 480 MB of RAM memory) by means of the xPC Target^{TM} toolbox of Matlab^{TM}.

In order to evaluate the vibration phenomena of the robot during the movements of its arms induced by the planned trajectories, a uni-axial accelerometer is mounted on the end-effector. The device has a maximum value of acceleration of ±5g and an accuracy of 1036 mV/g.

It is important to emphasize the fact that the evaluation of the vibration phenomena is only focused on the performance of the four trajectory planning algorithms, since the performances of the real time controller are not considered as fundamentals for the experimental tests. Starting from this assumption, the only inputs that can be changed in a “simulated” industrial task are the trajectories parameters, in good accordance with the conditions found in industrial environments, where generally a user is not allowed to change the parameters of the machine controller.

Table VIII : Kinematic bounds of the Cartesian manipulator

Joint	Kinematic Bounds		
	Velocity [mm/s]	Acceleration [mm/s^2]	Jerk [mm/s^3]
1	225	700	2400
2	225	700	2400
3	225	700	2400

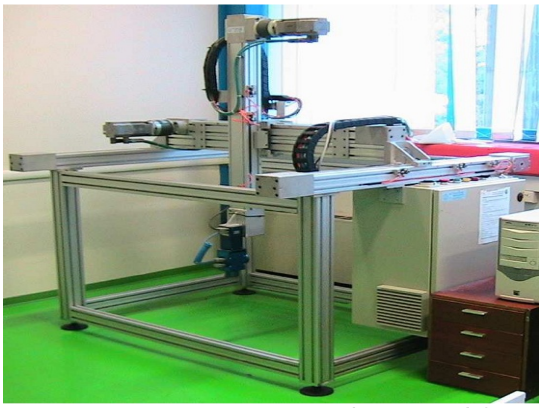

Figure 1: Cartesian manipulator used for testing the trajectory planning algorithms

❖ EVALUATION OF THE TRAJECTORIES SMOOTHNESS

The smoothness of the three trajectories planned with the four algorithms is experimental tested by means an accelerometer mounted on the robot end-effector. The direction used to measure the vibration of the manipulator has been chosen by taking into account the mean values of the simulated jerk along the path. By considering this assumption and the Tables III, V and VII, the X cartesian direction has been chosen for trajectories #2 and #3, whereas the Z cartesian direction has been chosen for trajectory #1.

Table IX : Measured accelerations mean value

	Accelerations mean value [m/s^2]		
	SPL3J	BSPL5J	GMJ
Trajectory #1	0.12	0.12	0.13
Trajectory #2	0.40	0.43	0.37
Trajectory #3	0.48	0.55	0.47

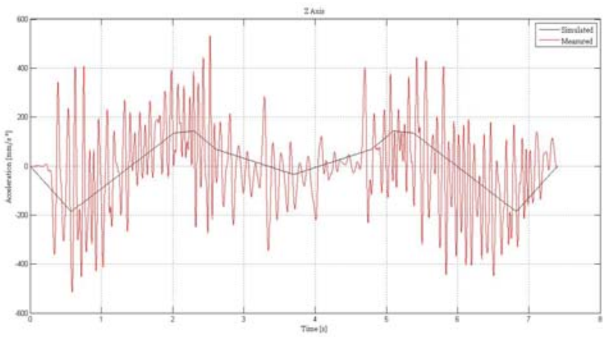

Figure 2 : Simulated vs. measured acceleration (Trajectory #1 - SPL3J)

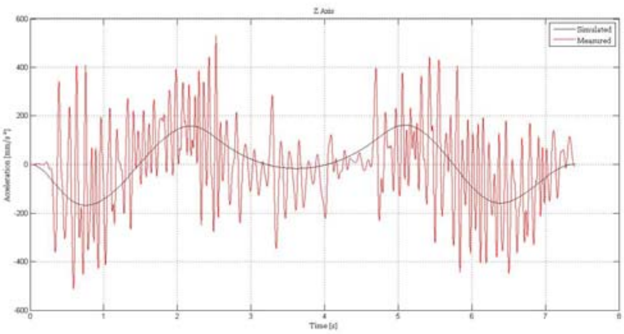

Figure 3 : Simulated vs. measured acceleration (Trajectory #1 - BSPL5J)

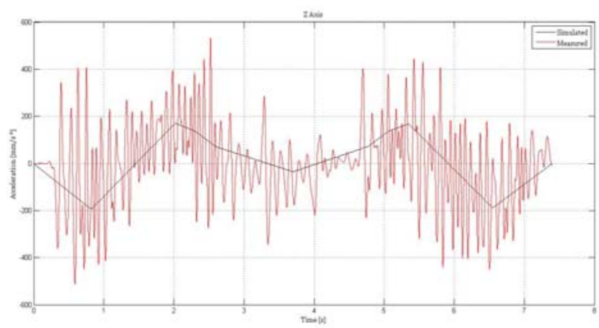

Figure 4 : Simulated vs. measured acceleration (Trajectory #1 - GMJ)

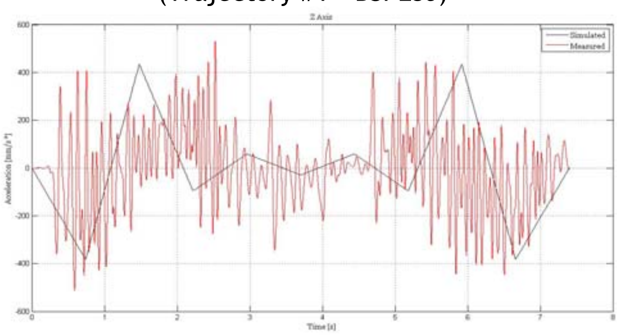

Figure 5 : Simulated vs. measured acceleration (Trajectory #1 - PROP)

In Table IX the mean values of the measured accelerations are reported. If the PROP values are considered as reference, a mean improvement of 36% is obtained if SPL3J and GMJ techniques are considered, a mean improvement of 31% is obtained if BSPL5J algorithm is used.

The comparison between the four trajectory planning algorithms for all the paths implemented, confirms the effectiveness of the SPL3J, BSPL5J and GMJ techniques in reducing the vibrations if compared to the PROP method.

All the experimental tests demonstrate that the real behavior of the Cartesian manipulator is effectively represented by the simulations, since the simulated accelerations obtained by running the algorithms and input to the manipulator have a time course comparable with the accelerations measured by the accelerometer mounted on the end-effector. To confirm this, in Figures 2-5 a comparison between the simulated and the measured accelerations (for Trajectory #1) is reported.

❖ CONCLUSION

In the present paper a minimum time-jerk trajectory planning technique has been experimental evaluated and validated. This method that takes into account both the integral of the squared jerk along the trajectory and its execution time, is implemented by using two types of primitives: cubic splines (SPL3J) and fifth-order B-splines (BSPL5J). The kinematic constraints are considered in the optimization problem, and the execution time is not set *a priori*.

An accelerometer mounted on the robot end-effector has been used with the aim to measure the accelerations of the manipulator joints, in order to evaluate the vibration phenomena of the Cartesian robot. Three test-trajectories have been implemented on a Cartesian manipulator and the experimental results have been compared with the results obtained with a global minimum jerk (GMJ) method, one of the most popular for planning smooth trajectories, and with a "classic" spline algorithm. The outcomes of the tests demonstrate the effectiveness of the smooth trajectory planning techniques, since the results prove the reductions of the vibration phenomena of the robot arms during the trajectory execution.

❖ REFERENCES

[1.] T. BROGARDH, Present and future robot control development – An industrial perspective, Annual Reviews in Control, 31 (1) (2007), 69-79.

[2.] A. PIAZZI and A. VISIOLI, Global minimum-time trajectory planning of mechanical manipulators using interval analysis, International Journal of Control, 71 (4) (1998), 631–652.

[3.] K. JOONYOUNG, K. SUNG-RAK, K. SOO-JONG, K. DONG-HYEOK, A practical approach for minimum-time trajectory planning for industrial robots, Industrial Robots : An International Journal, 37 (1) (2010), 51–61.

[4.] G. FIELD and Y. STEPANENKO, Iterative dynamic programming: an approach to minimum energy trajectory planning for robotic manipulators, Proc. of the IEEE International Conference on Robotics and Automation, 3 (1996), 2755–2760.

[5.] A. PIAZZI, A. VISIOLI, An interval algorithm for minimum-jerk trajectory planning of robot manipulators, Proceedings of the 36th Conference on Decision and Control, (1997), 1924–1927.

[6.] A. PIAZZI, A. VISIOLI, Global minimum-jerk trajectory planning of robot manipulators, IEEE Transactions on Industrial Electronics, 47 (1) (2000), 140–149.

[7.] H.XU, J.ZHUANG, S.WANG and Z.ZHU, Global Time-Energy Optimal Planning of Robot Trajectories, Proc. of the International Conference on Mechatronics and Automation, (2009), 4034–4039.

[8.] D. SIMON and C. ISIK, A trigonometric trajectory generator for robotic arms, International Journal of Control, 57 (3) (1993), 505–517.

[9.] P. HUANG, Y. XU and B. LIANG, Global minimum-jerk trajectory planning of space manipulator, International Journal of Control, Automation and Systems, 4 (4) (2006), 405–413.

[10.] K. PETRINEC and Z. KOVACIC, Trajectory planning algorithm based on the continuity of jerk, Proc. of the 15th Mediterranean Conference on Control & Automation, (2007).

[11.] A. GASPARETTO and V. ZANOTTO, A new method for smooth trajectory planning of robot manipulators, Mechanism and Machine Theory, 42 (4) (2007), 455–471.

[12.] A. GASPARETTO and V. ZANOTTO, A technique for time-jerk optimal planning of robot trajectories, Robotics and Computer-Integrated Manufacturing, 24 (3) (2008), 415–426.

[13.] F. LOMBAI and G. SZEDERKENYI, Throwing motion generation using nonlinear optimization on a 6-degree-of-freedom robot manipulator, Proc. of IEEE International Conference on Mechatronics, (2009).

[14.] A. GASPARETTO, A. LANZUTTI, R. VIDONI and V. ZANOTTO, Trajectory planning for manufacturing robots: algorithm definition and experimental results, Proc. of ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis ESDA2010, (2010).