A JERK-OPTIMAL MOTION PLANNING METHOD FOR ROBOTIC SYSTEMS

Florinel LUPU[1*], Sorin NEGROIU[2]

The paper presents an original method for jerk-optimal motion planning in robotic systems, where smooth motion and controlled dynamic loads are essential. By minimizing the overall jerk, the proposed approach enhances dynamic performance, reduces mechanical wear, and increases operational safety. Motion profiles are generated using cubic spline interpolation while satisfying position and velocity constraints. A numerical case study was conducted to compare the proposed method with a variational approach and with a numerical optimization technique. The results demonstrate the advantages of the method for real-time applications involving heavy-load manipulation.

Keywords: jerk minimization, cubic splines, motion smoothness, matrix method.

1. Introduction

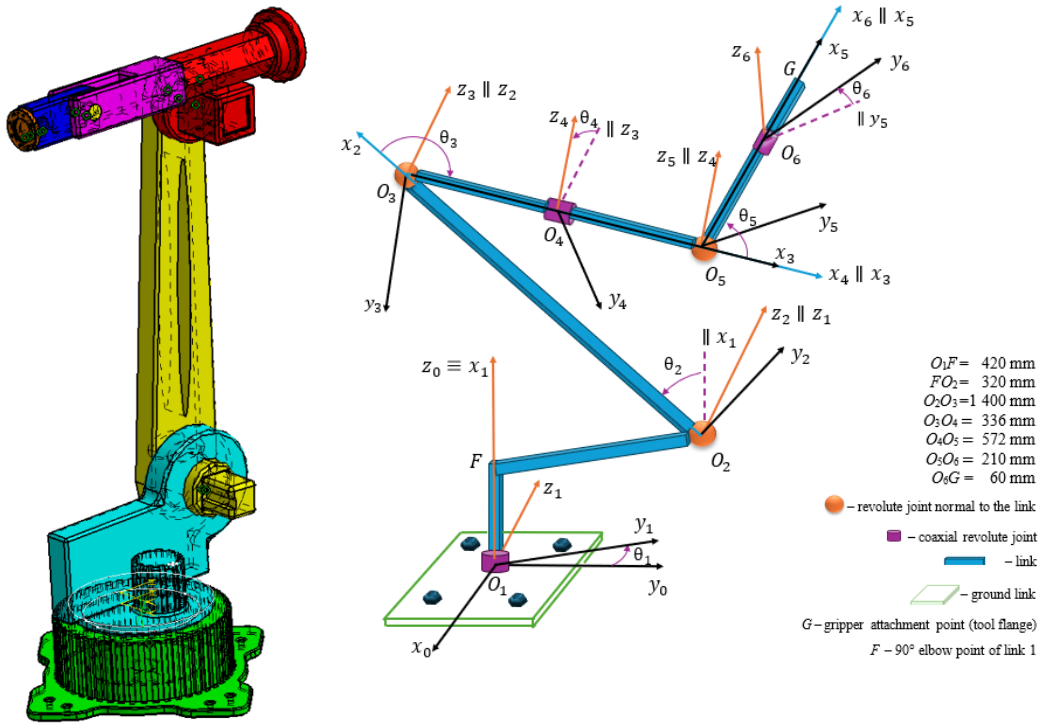
Motion planning is a key aspect of modern industrial robotics, particularly in applications involving complex tasks or the manipulation of heavy or irregularly shaped objects. While traditional approaches focused mainly on minimizing motion duration or ensuring positional accuracy, recent research emphasizes dynamic smoothness, especially jerk minimization, which directly affects mechanical stress, vibration levels, and the overall lifespan of robotic systems [1], [2]. Significant contributions in this field were made by Piazzi and Visioli [3], and by Gasparetto and Zanotto [4], [5], who proposed time–jerk optimal planning techniques based on spline interpolation under position and velocity constraints. Subsequent studies extended these ideas to more complex systems [6], flexible-joint robots [7], and high-precision manipulation tasks [8]. However, applications involving heavy load handling with long and rigid components remain challenging due to increased inertial forces and the risk of dynamic oscillations. The goal of this study is to develop a new jerk-optimal motion planning technique for a generic revolute joint of a robotic system. This work extends the authors' previous research [9], [10] by introducing an original method that minimizes the global jerk effect for a given operation time and leads to solving a linear algebraic system easily implementable in robot control. Compared to numerical conditional minimization methods such as

[1*] Eng., PhD student, National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: autokingdomlupu@gmail.com (corresponding author)
[2] Eng., PhD student, National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: sorinnegroiu458@gmail.com

those in [4], [5], the proposed approach provides higher accuracy. The method was implemented in MATLAB, and simulation results confirm its effectiveness for heavy-industry robotic applications.

2. Model of the robotic arm

The present study considers a six-axis industrial robot. Its geometry was modeled in Dassault Systèmes CATIA (Fig. 1a), where the complete 3D structure of the robot was constructed to accurately reflect the real mechanical configuration. The links were assigned distinct colors to clearly differentiate the structural components and joint interfaces, facilitating the interpretation of the kinematic chain and supporting the subsequent motion-planning analysis. The physical properties of the links (masses, centers of gravity, volumes, and inertial data) extracted from the CATIA model are provided in Appendix A. The kinematic model (Fig. 1b) illustrates the joint configuration through angles θ_1, θ_2, …, θ_6. Motion generation consists in determining the time evolution of these parameters. In the following, an optimal motion generation method is developed, and for clarity the generic notation q is used to denote any joint angle θ_j from the kinematic model.



a) 3D model assembly designed in CATIA

b) Kinematic model of the robot

Fig. 1. 6-axis industrial robot

3. Motion planning method

The motion generation is based on a modified version of the method proposed by Gasparetto and Zanotto, which employs cubic spline functions to ensure continuity of position, velocity, and acceleration.

In the following, q_i is the position parameter (rotation angle of a revolute joint) at time moment t_i, with $i = 1, \dots, n$. On each time interval $[t_i, t_{i+1}]$ a cubic spline approximation function is defined as [4], [5]:

$$Q_i(t) = \frac{\alpha_i}{6h_i}(t_{i+1} - t)^3 + \frac{\alpha_{i+1}}{6h_i}(t - t_i)^3 + \left(\frac{q_{i+1}}{h_i} - \frac{h_i\alpha_{i+1}}{6}\right)(t - t_i) + \left(\frac{q_i}{h_i} - \frac{h_i\alpha_i}{6}\right)(t_{i+1} - t), \quad (1)$$

where

$h_i = t_{i+1} - t_i$ is the time sampling step,

$\alpha_i = \ddot{Q}_i$ is the angular acceleration at time moment t_i,

$Q_i(t)$ is the interpolated position parameter in the interval $[t_i, t_{i+1}]$.

The first derivative of expression (1) is the joint angular velocity:

$$\dot{Q}_i(t) = -\frac{\alpha_i}{2h_i}(t_{i+1} - t)^2 + \frac{\alpha_{i+1}}{2h_i}(t - t_i)^2 + \left(\frac{q_{i+1} - q_i}{h_i}\right)\frac{h_i}{6}(\alpha_{i+1} + \alpha_i)\,. \quad (2)$$

The joint angular acceleration is:

$$\ddot{Q}_i(t) = \frac{t_{i+1} - t}{h_i}\alpha_i + \frac{t - t_i}{h_i}\alpha_{i+1}. \quad (3)$$

The joint angular jerk is:

$$\dddot{Q}_i(t) = \frac{\alpha_{i+1} - \alpha_i}{h_i}. \quad (4)$$

To ensure smoothness, the angular velocity must be continuous at internal nodes:

$$\dot{Q}_i(t_{i+1}) = \dot{Q}_{i+1}(t_{i+1}) \quad (i = 1, \dots n-2). \quad (5)$$

The limit conditions in position are implicitly satisfied by choosing form (1), while limit conditions in angular velocity imply the equalities

$$\begin{cases} \dot{Q}_1(t_1) = \dot{q}_1 \\ \dot{Q}_{n-1}(t_n) = \dot{q}_n\,. \end{cases} \quad (6)$$

Relations (5) and (6) lead to a linear system in the unknown angular accelerations α_i, which can be written in matrix form as

$$[K]\{\alpha\} = \{B\}, \quad (7)$$

where

$$[K]=\begin{bmatrix} \frac{h_1}{3} & \frac{h_1}{6} & 0 & \dots & 0 & 0 \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & \dots & 0 & 0 \\ 0 & \frac{h_2}{6} & \frac{h_2+h_3}{3} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \frac{h_{n-2}+h_{n-1}}{3} & \frac{h_{n-1}}{6} \\ 0 & 0 & 0 & \dots & \frac{h_{n-1}}{6} & \frac{h_{n-1}}{3} \end{bmatrix} \tag{8}$$

is a tridiagonal n-th-order square matrix,

$$\{\alpha\}=(\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_n)^T \tag{9}$$

is the column matrix of the unknown joint angular accelerations, while

$$\{B\}=\begin{Bmatrix} -\dot{q}_1+\frac{q_2-q_1}{h_1} \\ \frac{q_1}{h_1}-\left(\frac{1}{h_1}+\frac{1}{h_2}\right)q_2+\frac{q_3}{h_2} \\ \frac{q_2}{h_2}-\left(\frac{1}{h_2}+\frac{1}{h_3}\right)q_3+\frac{q_4}{h_3} \\ \dots \\ \frac{q_{n-2}}{h_{n-2}}-\left(\frac{1}{h_{n-2}}+\frac{1}{h_{n-1}}\right)q_{n-1}+\frac{q_n}{h_{n-1}} \\ \dot{q}_n+\frac{q_{n-1}-q_n}{h_{n-1}} \end{Bmatrix} \tag{10}$$

is a column matrix, dependent on the joint position parameters and on the limit angular velocities.

After solving the linear system (7), angular accelerations α_i are used in formulae (1)–(4) to approximate the position parameter, angular velocity, acceleration and jerk profiles.

4. Global jerk effect parameter

The global jerk effect is expressed by the parameter [4], [5]

$$j=\int_{t_1}^{t_n}\dddot{q}^2dt\,. \tag{11}$$

Approximating q by the spline interpolation Q, parameter j can be written

$$j\approx J(\alpha)=\int_{t_1}^{t_n}\dddot{Q}^2dt=\sum_{i=1}^{n-1}\frac{(\alpha_{i+1}-\alpha_i)^2}{h_i^2}\cdot h_i=(\alpha)[E]\{\alpha\}\,, \tag{12}$$

where

$$[E]=\begin{bmatrix} \frac{1}{h_1} & -\frac{1}{h_1} & 0 & \dots & 0 & 0 \\ -\frac{1}{h_1} & \frac{1}{h_1}+\frac{1}{h_2} & -\frac{1}{h_2} & \dots & 0 & 0 \\ 0 & -\frac{1}{h_2} & \frac{1}{h_2}+\frac{1}{h_3} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \frac{1}{h_{n-2}}+\frac{1}{h_{n-1}} & -\frac{1}{h_{n-1}} \\ 0 & 0 & 0 & \dots & -\frac{1}{h_{n-1}} & \frac{1}{h_{n-1}} \end{bmatrix}, \tag{13}$$

$$(\alpha)=\{\alpha\}^T. \tag{14}$$

5. Original motion-optimization method

In references [4] and [5], Gasparetto and Zanotto present an optimization method aimed at reducing both the operation time and the global jerk effect. Their technique involves determining the time grid that minimizes an objective function for a given trajectory in the configuration space, and it relies on relatively complex numerical computations.

The present paper adopts a different strategy by determining the motion of a generic revolute joint that minimizes the global jerk effect for a given time grid and specified limit conditions.

Column matrix $\{B\}$ can be expressed as

$$\{B\}=\{B\}_0+[B]\{q\}, \tag{15}$$

where

$$\{B\}_0=(-\dot{q}_1 \quad 0 \quad \dots \quad 0 \quad \dot{q}_n)^T, \tag{16}$$

$$\{q\}=(q_1 \quad q_2 \quad \cdots \quad q_n)^T, \tag{17}$$

$$[B]=\begin{bmatrix} -\frac{1}{h_1} & \frac{1}{h_1} & 0 & \dots & 0 & 0 \\ \frac{1}{h_1} & -\left(\frac{1}{h_1}+\frac{1}{h_2}\right) & \frac{1}{h_2} & \dots & 0 & 0 \\ 0 & \frac{1}{h_2} & -\left(\frac{1}{h_2}+\frac{1}{h_3}\right) & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -\left(\frac{1}{h_{n-2}}+\frac{1}{h_{n-1}}\right) & \frac{1}{h_{n-1}} \\ 0 & 0 & 0 & \dots & \frac{1}{h_{n-1}} & -\frac{1}{h_{n-1}} \end{bmatrix}=-[E] \tag{18}$$

or

$$\{B\} = \{B\}_0^* + [B]^*\{q\}^*, \tag{19}$$

where

$$\{q\}^* = (q_2 \quad q_3 \quad \cdots \quad q_{n-1})^T, \tag{20}$$

$$\{B\}_0^* = \begin{Bmatrix} -\frac{1}{h_1}q_1 - \dot{q}_1 \\ \frac{1}{h_1}q_1 \\ 0 \\ \cdots \\ \frac{1}{h_{n-1}}q_n \\ -\frac{1}{h_{n-1}}q_n + \dot{q}_n \end{Bmatrix} q_n, \tag{21}$$

$$[B]^* = \begin{bmatrix} \frac{1}{h_1} & 0 & \cdots & 0 \\ -\left(\frac{1}{h_1}+\frac{1}{h_2}\right) & \frac{1}{h_2} & \cdots & 0 \\ \frac{1}{h_2} & -\left(\frac{1}{h_2}+\frac{1}{h_3}\right) & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -\left(\frac{1}{h_{n-2}}+\frac{1}{h_{n-1}}\right) \\ 0 & 0 & \cdots & \frac{1}{h_{n-1}} \end{bmatrix}. \tag{22}$$

Using relation (19), the solution of system (7) becomes:

$$\{\alpha\} = [K]^{-1}\{B\}_0^* + [K]^{-1}[B]^*\{q\}^*. \tag{23}$$

By transposing equality (23), the following is obtained:

$$(\alpha) = \{\alpha\}^T = (B)_0^*[K]^{-1^T} + (q)^*[B]^{*^T}[K]^{-1^T}. \tag{24}$$

Using (23) and (24), function (12) takes the form

$$J(q^*) = J\big(\alpha(q^*)\big) = (q)^*[G]\{q\}^* + 2(H)\{q\}^* + f_0\,, \tag{25}$$

where:

$$[G] = [B]^{*^T}[K]^{-1^T}[E][K]^{-1}[B]^*, \tag{26}$$

$$(H) = (B)_0^*[K]^{-1^T}[E][K]^{-1}[B]^*, \tag{27}$$

$$f_0 = (B)_0^*[K]^{-1^T}[E][K]^{-1}\{B\}_0^* . \tag{28}$$

The optimality condition consists in determining the values of parameters q_2, q_3,…, q_{n-1} for which the function J attains its minimum. This occurs when the partial derivatives of the function are equal to zero:

$$\frac{\partial J}{\partial q_i} = 0 \quad (i = 2,3,\dots,n-1) . \tag{29}$$

By replacing the expressions of the partial derivatives of function J, system (29) takes the matrix form

$$[G]\{q\}^* + \{H\} = \{0\} . \tag{30}$$

The solution of this system is

$$\{q\}^*_{optim} = -[G]^{-1}\{H\} . \tag{31}$$

6. Classical variational approach

A classical approach consists in determining the exact optimal motion using the methods of the variational calculus. This solution can be used as a standard reference.

In course [11] it is shown that the extreme of a functional of the form

$$J = \int_{t_1}^{t_n} F(t, q, \dot{q}, \ddot{q}, \dddot{q})\, d\, t \tag{32}$$

is determined by the differential equation

$$\frac{\partial F}{\partial q} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial F}{\partial \dot{q}}\right) + \frac{\mathrm{d}^2}{\mathrm{d}t^2}\left(\frac{\partial F}{\partial \ddot{q}}\right) - \frac{\mathrm{d}^3}{\mathrm{d}t^3}\left(\frac{\partial F}{\partial \dddot{q}}\right) = 0 . \tag{33}$$

As follows from (11), in the studied application, function F is

$$F(t, q, \dot{q}, \ddot{q}, \dddot{q}) = \dddot{q}^2 . \tag{34}$$

By substituting (34) in (33), the following differential equation is obtained:

$$\frac{\mathrm{d}^3\dddot{q}}{\mathrm{d}\, t^3} = 0 . \tag{35}$$

The solution of equation (35) is a fifth-order polynomial:

$$\begin{aligned} q(t) = a_5(t-t_1)^5 + a_4(t-t_1)^4 + a_3(t-t_1)^3 + a_2(t-t_1)^2 \\ + a_1(t-t_1) + a_0 . \end{aligned} \tag{36}$$

The coefficients of the polynomial (36) can be determined from the limit conditions. Four such conditions are expressed in terms of position and velocity,

$$\begin{cases} q(t_1) = q_1 \\ q(t_n) = q_n \\ \dot{q}(t_1) = 0 \\ \dot{q}(t_n) = 0\,, \end{cases} \tag{37}$$

while two others are obtained from natural limit conditions applied to accelerations:

$$\begin{cases} \ddot{q}(t_1) = ? \quad \text{so that} \quad J = \min \\ \ddot{q}(t_n) = ? \quad \text{so that} \quad J = \min\,. \end{cases} \tag{38}$$

Following the procedure in [11], the last two conditions can be expressed as

$$\begin{cases} \left.\dfrac{\partial F}{\partial \ddot{q}}\right|_{t=t_0} = 0 \\ \left.\dfrac{\partial F}{\partial \ddot{q}}\right|_{t=t_1} = 0\,, \end{cases} \tag{39}$$

which lead to

$$\begin{cases} \ddot{q}(t_1) = 0 \\ \ddot{q}(t_n) = 0\,. \end{cases} \tag{40}$$

Using conditions (37) and (40), polynomial (36) takes the form:

$$q(t) = (q_n - q_1)\left[\frac{(t-t_1)^5}{(t_n-t_1)^5} - \frac{5}{2}\frac{(t-t_1)^4}{(t_n-t_1)^4} + \frac{5}{2}\frac{(t-t_1)^2}{(t_n-t_1)^2}\right] + q_1. \tag{41}$$

Replacing expression (41) in formula (11), the following is obtained:

$$J = 120\frac{(q_n-q_1)^2}{(t_n-t_1)^5}\,. \tag{42}$$

7. Numerical application

A two-stage numerical application was developed in MATLAB, in order to assess the accuracy of several motion generation methods, compare the corresponding optimization techniques, and evaluate the global jerk effect. The test motion starts and ends at rest, with the input data $t_1 = 0$, $t_n = 2$ s, $q_1 = 0.2$ rad, $q_n = 0.8$ rad, $\dot{q}_1 = 0$, and $\dot{q}_n = 0$. Computations were performed for sampling steps $h_i = 0.2$ s, 0.1 s, 0.04 s, and 0.02 s, while the interpolation step was set to $\Delta t = 0.1h_i$ in all cases, ensuring uniform refinement. The MATLAB scripts used to generate trajectories and figures are available upon request from the editorial board.

7.1. Assessment of the accuracy of various motion generation methods

The motion was generated based on four different laws, as described below.
a) First, a trigonometrical variation was considered (Fig. 2 – red curves)

$$q(t) = q_1 + \frac{q_n - q_1}{2}\left(1 - \cos \pi \frac{t - t_1}{t_n - t_1}\right), \tag{43}$$

Replacing expression (43) in formula (11), the following is obtained:

$$j = \frac{\pi^6}{8}\frac{(q_n - q_1)^2}{(t_n - t_1)^5} = 120.17\frac{(q_n - q_1)^2}{(t_n - t_1)^5}. \tag{44}$$

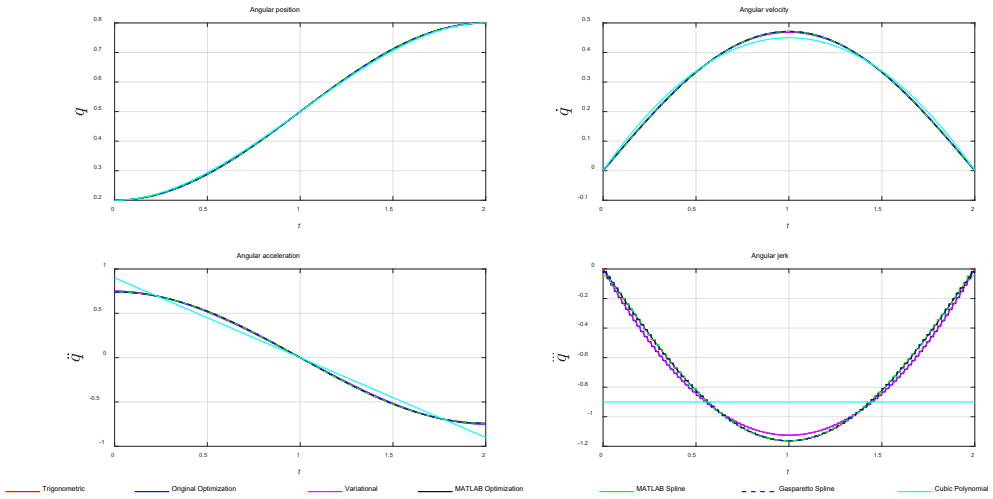


Fig. 2. Variations of four kinematic parameters ($h_i = 0.02$ s, $\Delta t = 0.002$ s)

b) A second variation (Fig. 2 – green curves) was generated using spline interpolation, implemented through the MATLAB function *spline*, determined from samples q_1, q_2,…, q_n, defined by function (43). This case was introduced to assess the quality of the spline approximation of the original trigonometric function. It should be noted that function *spline* enforces continuity of velocities (5) but, instead of limit conditions (6), it assumes equal values of jerk on the first two subintervals, $[t_1, t_2]$ and $[t_2, t_3]$, as well as on the last two ones, $[t_{n-2}, t_{n-1}]$ and $[t_{n-1}, t_n]$.

c) The third variation (Fig. 2 – dark blue curves) was also generated using spline interpolation, this time following the method of Gasparetto and Zanotto, determined using the same samples defined by function (43). Unlike MATLAB interpolation, in this case limit conditions (6) were applied, which are more realistic, since motions of robotic elements usually start and end at rest.

d) Finally, a fourth variation (Fig. 2 – cyan curves) was examined, assuming a cubic polynomial curve for the position parameter over the entire operation time interval:

$$q(t) = (q_n - q_1)\left[-2\frac{(t-t_1)^3}{(t_n-t_1)^3} + 3\frac{(t-t_1)^2}{(t_n-t_1)^2}\right] + q_1 \,. \tag{45}$$

This case was introduced to investigate the behavior of a simplified approximation of the original trigonometric function.

Replacing expression (45) in formula (11), the following is obtained:

$$j = 144\frac{(q_n-q_1)^2}{(t_n-t_1)^5}\,. \tag{46}$$

7.2. Comparative analysis of the optimization methods

To compare the advantages and limitations of the optimization methods under study, four different motion laws were analyzed, as follows.

a) The non-optimized trigonometric variation (43) (Fig. 2 – red curves), which was considered as a reference.

b) A variation generated using the original method presented in section 5 (Fig. 2 – dark blue curves).

c) A motion obtained using the variational-calculus-optimization described in section 6 (Fig. 2 – magenta curves).

d) A MATLAB optimization, using *fminsearch*, with respect to the variables q_i (for $i = 2,3,\dots,n-1$), assuming known values of t_i (Fig. 2 – black curves).

7.3. Evaluation of the global jerk effect parameter

The global jerk effect parameter (j for the analytical variations and J for the numerical ones) was computed in MATLAB and is presented in Table 1, associating each method with the corresponding formula.

Table 1

***j/J* [rad²/s⁵]**

Method	Formula	h_i = 0.2s	h_i = 0.1s	h_i = 0.04s	h_i = 0.02s
Trigonometric variation	(44)	1.3520	1.3520	1.3520	1.3520
Cubic polynomial variation	(46)	1.6200	1.6200	1.6200	1.6200
MATLAB spline curve	(12)	1.4140	1.3615	1.3528	1.3521
Gasparetto and Zanotto spline	(12)	1.3631	1.3547	1.3524	1.3521
Original optimization	(12)	1.3613	1.3528	1.3505	1.3501
MATLAB optimization	(12)	1.3629	1.3547	1.3524	1.3521
Variational optimization	(42)	1.3500	1.3500	1.3500	1.3500

The results show that all numerically generated motions progressively converge toward the analytical jerk value as the sampling step decreases. Interpolation-based motions approach the analytical reference rapidly, while optimization-based motions achieve the lowest global jerk values for the finest discretization. Figure 3 confirms this trend, illustrating smoother profiles and reduced differences between methods at smaller h_i.

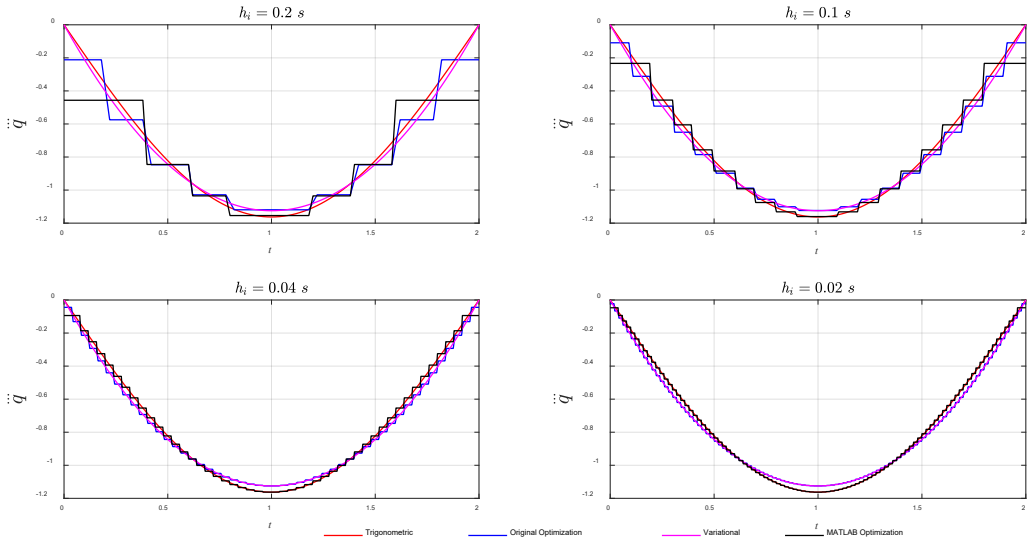


Fig. 3. Dependence of variation of jerk on the time sampling step h_i

8. Conclusions

The motion planning of a generic revolute joint of a robotic system was achieved by approximating the variation of the angular position parameter using cubic spline functions, which ensure continuous position, velocity, and acceleration profiles over the entire operation interval. Based on this approximation method, an optimization problem was formulated as the minimization of an integral performance index with respect to the position values at the spline nodes.

The analysis of the motion generation methods (Section 7.1) shows that spline-based approximations converge toward the analytical curves as the time sampling step decreases. Although the position, angular velocity, and angular acceleration obtained through spline interpolation remain close to the analytical reference, their higher-order derivatives exhibit progressively larger deviations, with jerk being the most sensitive quantity, as illustrated in Figure 2. Among the tested motion laws, the cubic polynomial produces the largest deviations relative to the trigonometric reference, making it unsuitable for accurate modeling.

An original optimization method was developed, and the resulting law of motion was compared with the non-optimized trigonometric law, with the exact solution obtained through variational calculus, and with the solution obtained by numerical optimization using the MATLAB *fminsearch* function. As shown in Figures 2 and 3, for all analyzed sampling steps, the original method provides a closer match to the exact variational solution than the MATLAB-based optimization. Furthermore, as the sampling step decreases, the original optimization method converges toward the exact variational solution, whereas the numerically optimized one remains close to the non-optimized trigonometric form. This confirms that the proposed method achieves higher accuracy while maintaining computational simplicity..

The computation of the global jerk effect parameter, j or J, summarized in Table 1, further clarifies the behavior of the optimization techniques. Both the original and the MATLAB-based methods become unreliable for coarse sampling steps. Even though each step is subdivided into ten equal subintervals for interpolation, such discretization still leads to noticeable degradation in motion quality, with the resulting jerk effect parameter exceeding that of the non-optimized trajectories. The cubic polynomial variation over the entire operation interval produces the highest jerk effect parameter overall, making it the least suitable option for applications requiring smooth motion profiles.

Future work may explore joint-space multiobjective optimization, considering both jerk and energy, as well as alternative motion laws capable of generating smoother transitions and improved dynamic performance in robotic tasks. A further direction will be the extension of the optimization method introduced in this work to minimize the jerk of each joint, weighting their contributions according to dynamic sensitivity, thereby reducing the overall jerk of the robot. This extension will enable the method, originally formulated for a generic revolute joint, to be applied to the complete six-joint robotic model represented in CATIA.

R E F E R E N C E S

[1] *N. Hogan*, Impedance control: An approach to manipulation, ASME Journal of Dynamic Systems, Measurement, and Control, **107**(1), 1985, pp. 1–24.

[2] *J. Angeles*, Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms, Springer, New York, 2007.

[3] *A. Piazzi and A. Visioli*, Global minimum-time trajectory planning of mechanical manipulators using interval analysis, International Journal of Control, vol. 71, no. 4, pp. 631–652, 1998.

[4] *A. Gasparetto*, *V. Zanotto*, A technique for time-jerk optimal planning of robot trajectories, Robotics and Computer-Integrated Manufacturing, **26**(3), 2010, pp. 361–371.

[5] *A. Gasparetto*, *V. Zanotto*, A new method for smooth trajectory planning of robot manipulators, Mechanism and Machine Theory **42**, 2007, pp. 455–471.

[6] *D. Kulic*, *E. Croft*, Safe planning for human-robot interaction, Journal of Robotic Systems, **22**(7), 2005, pp. 383–396.

[7] *B. Siciliano*, *L. Sciavicco*, *L. Villani*, *G. Oriolo*, Robotics: Modelling, Planning and Control, Springer, London, 2009.
[8] *H. Asada*, *J.-J. E. Slotine*, Robot Analysis and Control, John Wiley & Sons, New York, 1986.
[9] *F. Lupu, A. Craifaleanu,* Kinematic Study of a robotic component with relative motion, International Symposium ISB-INMA TEH', Bucharest, 2024, pp. 1214-1221.
[10] *F. Lupu, A. Craifaleanu, S. Negroiu*, Study of jerk for a robotic arm, International Symposium ISB-INMA TEH', Bucharest, 2025, pp. 306-311.
[11] *V. I. Smirnov,* Cours de mathématiques supérieures, tome IV deuxième partie, Editions Mir, Moscou, 1969-1984.

Appendix A

Physical properties of the robot components

The physical properties listed in this appendix were computed assuming a steel density of $\rho = 7.85$ g/cm³, which was used to determine the mass, center of gravity, and inertia values of all robot components.

1. Mass and volume

Component	Mass [g]	Volume [mm³]
Base	51 324.555	51 324 555.365
Link 1	497 147.552	63 250 324.647
Link 2	381 067.194	48 481 831.361
Link 3	168 279.110	21 409 556.026
Link 4	70 517.414	8 971 681.184
Link 5 (Assembly)	73 632.426	9 367 993.564
Link 6	3 810.226	484 761.592
Total	**1 198 478**	**203 290 703.739**

2. Center of Gravity

Component	Gx [mm]	Gy [mm]	Gz [mm]
Base	0.031	−0.248	−67.663
Link 1	35.658	13.715	−293.941
Link 2	339.805	159.114	−1 140.899
Link 3	284.350	344.144	−2 054.813
Link 4	−5.762	105.893	−2 151.378
Link 5 (Assembly)	−13.08	101.44	−2 151.97
Link 6	−377.623	−251.623	−2 170.703

3. Moments of Inertia About the Center of Gravity

Component	IoxG [kg·m²]	IoyG [kg·m²]	IozG [kg·m²]
Link 1	25.209	26.758	18.602
Link 2	94.599	95.029	4.189
Link 3	3.605	3.273	3.767
Link 4	1.216	1.274	2.227
Link 5 (Assembly)	1.268	1.326	2.279
Link 6	0.007	0.007	0.008