



Optimal trajectory generation method to find a smooth robot joint trajectory based on multiquadric radial basis functions

Bendali Nadir^{1,2} · Ouali Mohammed² · Nguyen Minh-Tuan³ · Said Abderrezak¹

Received: 1 September 2021 / Accepted: 6 January 2022 / Published online: 31 January 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

A new technique to generate smooth motion trajectories for robot manipulators using multiquadric radial basis functions (MQ-RBFs) is presented in this paper. In order to get the optimal trajectory, two objective functions are minimized that are proportional to the execution time, the integral of the squared jerk (which denotes the time derivative of the acceleration) along the whole trajectory. Also, the proposed interpolation technique is introduced for solving the trajectory planning problem in the joint space, where the interpolation of via-points takes into account boundary conditions and also satisfies kinematics limits of velocity, acceleration, and jerk. Then, the proposed approach is compared with a set of classical interpolation techniques based on radial basis function models and cubic splines. Finally, the proposed technique has been tested for the six-joint PUMA 560 manipulator in two cases (minimum time and minimum time-jerk) and results are compared with those proposed of other important trajectory planning techniques. Numerical results show the competent performances of the proposed methodology to generate trajectories in short total transfer time and with high smooth profile.

Keywords MQ-RBF · Trajectory generation · Execution time · Jerk · Robotic manipulator

1 Introduction

Robot manipulators are mechanisms composed of an assembly of links and joints that is roughly analogous to the human arm; the end-effector is usually equipped with a gripping tool making it possible to manipulate objects. These robots are often intended for performing complex and repetitive tasks

in large workspaces. In order to enhance the efficiency of a robotic manipulator when performing a given operation, it is necessary to determine the best trajectory to execute this task by the optimization of certain variables.

Generally, the robot manipulators execute two types of tasks, namely, continuous tasks (CT) and pick-and-place tasks (PPT). In CT, the robot end-effector is meant to attend a smooth and continuous trajectory. This appears in operations such as flame-cutting, arc-welding, and deburring. In PPT, only the first and last configurations should be respected in addition to the obstacles to be avoided. These two configurations are connected with a free trajectory. This can be the case with placing components on circuit boards, point-to-point welding, and tool changes in machine tools [1].

As the robot can be described in two types of tasks, continuous tasks, and pick-and-place tasks, it is the same for the trajectory. Usually, trajectory planning is performed in the operational space (OS) and the joint space (JS). In OS, the trajectory can be naturally described by the end-effectors, and it is easy to display and visualize the trajectory. On the other hand, it is difficult to guarantee that singularities do not occur. Trajectory planning in the JS will reach the points defined in the OS with better accuracy since errors due to the solution of the forward and reverse kinematic models are not introduced in

✉ Bendali Nadir
n.bendali@univ-dbkm.dz

Ouali Mohammed
oualimohammed@yahoo.fr

Nguyen Minh-Tuan
tuanctm7@gmail.com

Said Abderrezak
a.said@univ-dbkm.dz

¹ Faculty of Sciences and Technology, University of Khemis Miliana, Road of Theniet El Had, Khemis Miliana 44225, Algeria

² Structural Mechanics Research Laboratory, Department of Mechanical Engineering, University Blida-1, Blida, Algeria

³ Hung Yen University of Technology and Education, Hung Yen, Vietnam

the system. However, in the majority of cases, the trajectory of the robots is planned in the joint space because, since the joints of the robot manipulators receive the action of the control, planning in the operational space requires a kinematic inversion in order to derive the evolution of variables in the JS [2].

High operating speed is one of the essential requirements to minimize production cycle time in many automated production systems, especially in robotic units. Thus, the development of a method that obtains a fast movement of robots has been broadly studied in the subject of robot manipulators. However, considering the high non-linearity of the systems and coupling between joints and the gravity effects of rigid links, it is possible to get approximate solutions [3, 4]. In this area, the authors proposed an important phase plane method [5, 6]. However, the torque and acceleration profiles of these algorithms present a discontinuity, and this is due that the links that constitute the robot are perfectly rigid and to the fact that the actuator dynamics are neglected. Later, to overcome this kind of drawback, the authors in [7] introduced actuator jerk constraint limits. However, their method could not be precisely optimal over time, but the trajectories generated might be used for further innovative control approaches. Xiao et al. [8] recently obtained minimum traveling time and smooth trajectory for online operations, converting kinematic and dynamic constraints of joint space and the path of operating space into parameter space by using the cubic polynomial fitting technique. The optimization problem of time-optimal becomes non-convex by considering the viscous friction of the mechanism. Other minimum time methods using genetic algorithms for kinematics and dynamics trajectory planning can be found in [9] and [10], respectively. Abu-Dakka et al. [11] established a new genetic algorithm method for optimal trajectory and without collision between the first and last configurations in complex environments as the robot moves.

Apart from optimization of traveling time, bounding or minimizing the jerk is necessary because low-jerk trajectories may be achieved more smoothly and precisely. Furthermore, a high-jerk value can heavily excite its resonance frequencies and wear out the robot structure; vibrations produced by non-smooth trajectories can present significant errors and damage the actuators while the robot is completing tasks as trajectory tracking [12]. For the optimization of the jerk, the value of jerk in the objective function can appear in two principal approaches, one as the maximum absolute value of the jerk (minimax approach) in studies [13–17], while in [18–20] as the time integral of the squared jerk along the trajectory for a robot manipulator.

In the manufacturing process, the movements of industrial robots are habitually needed to be performed in a time-jerk optimal way. Trajectory planning approaches along with continuous tasks (CT) with time-jerk optimal have also been proposed in scientific research [21–26]. Different optimization approaches were utilized to get the optimal trajectory by minimizing the hybrid objective function composed of two terms proportional to the traveling time and the interval squared jerk;

for example, studies in Refs. [23, 26, 27] used the sequential quadratic programming (SQP) technique, while in Ref. [24], a meta-heuristic approach, namely teaching learning based optimization (TLBO), was employed for producing time-optimal smooth motion trajectories. Another hybrid method is that in Refs. [28, 29], where a time energy smoothness trajectory planning algorithm is presented. In this technique, the trajectories are generated in third-order spline form using general constrained nonlinear optimization, considering the joint physical limits and the actuator's power consumption.

In robotics, trajectories are mostly described by algebraic splines (e.g., quintic splines [27], cubic splines [10, 30], quartic splines [31], B-splines [32–35], and trigonometric splines [18]), Bezier [36] and polynomial interpolation functions in [37, 38]. Sometimes a mix of them is established to produce the best reference trajectories. For example, in [39], the authors combine fourth- and fifth-order polynomials, while [40] combined 7th-order polynomials and cubic splines to guarantee zero jerk at the joint movement's limits. Ref. [41] proposed a smooth trajectory planning approach with the combination of septuple B-splines in JS and cubic splines in OS of robot manipulator.

In addition to these broad varieties of interpolation functions, particularly we pay attention to radial basis functions (RBFs), which are an advanced technique for scattered data interpolation based on linear combinations of terms that include a single univariate function [42]. However, to our knowledge, there are not many publications in scientific research dealing with the trajectory planning problems by using RBF interpolation methods for robotic systems. In this context, in Ref. [43] the authors presented a new approach for planning trajectory in pick-and-place operations using Gaussian RBF; the obtained trajectory is compared with those of other interpolation techniques. Reference [44] offers a direct solution to optimal control problems. It is based on the interpolation of global radial basis functions (RBFs) on arbitrary collocation points. In their proposed approach, states and controls are parameterized with any arbitrary global RBF and the continuous-time optimal control problem is discretized using arbitrary collocation points to translate it into a nonlinear programming problem, while in Ref. [45] combining radial basis function (RBF) interpolation with Galerkin projection provides a new approach to effectively solve general optimal control problems. They develop a very flexible solution to optimal control problems, especially non-smooth problems involving discontinuities, while simultaneously taking into account trajectory accuracy and computational efficiency, and Ref. [46] proposed to use radial basis functions with Gaussian kernels for generating smooth Cartesian paths for robot where the motion is defined by a sequence of intermediate knots through which the robot has to move.

This paper aims to propose an original trajectory planning technique using MQ-RBFs with the requirements of rapidity and high-precision operations for robotic systems. The practical interest of this study is that MQ-RBF can simply be used for planning

smooth trajectories with continuous derivatives for any order and with specific boundary conditions. Indeed, a robot manipulator is basically a precision machine in which vibrations are regarded as an unwanted phenomenon. These vibrations are caused by the presence of structural elasticity in the mechanical system, and they can be induced if the actuation devices are subjected to trajectories with a discontinuous acceleration profile. Furthermore, a considerable discontinuity in acceleration induces a rapid variation in the inertial forces applied to the robot structure, which will excite inherent structural elasticity. In consequence, this problem must be addressed during the trajectory planning process by selecting adequate profiles and imposing appropriate continuity and end conditions. Moreover, an important property of MQ-RBF is the existence of a shape parameter that significantly affects the profile for the generated trajectories of robot manipulators. Performance of the proposed approach was tested and verified for different examples in continuous tasks/pick-and-place tasks, with/without optimization and by using other interpolation functions in order to obtain an optimal trajectory.

The rest of the paper will be arranged as follows. Section 2 defines the problem of trajectory optimization, namely the objective function to minimize using sequential quadratic programming (SQP) techniques. In Sect. 3, multiquadric-RBF is adopted to model the joint trajectory, considering boundary conditions and the optimization process. Then, in Sect. 4, the proposed method is compared with a set of classical planning approaches. In Sect. 5, the proposed interpolation technique is tested to the PUMA 560 industrial robot with six revolute joints for two cases by minimizing: the first case is the transfer time and the second one is the bi-objective time-jerk index. Finally, the main conclusions are reported in Sect. 6.

2 Problem statement

In this problem, the PUMA 560 robot manipulator with six degrees of freedom is considered. In industrial operations, the robot manipulators must pass through a sequence of intermediate knots in OS. However, as described in the previous section, the execution of trajectory planning in the task space is relatively complicated, and the via-points are converted into JS using kinematic inversion. Hence, a trajectory will be constructed based on the via-points in JS and then minimized for some cost functions.

The robot manipulator’s trajectory is required to take as little transfer time as possible to increase productivity and be smooth enough to avoid excessive mechanical vibrations by applying kinematic constraints.

Hence, the optimization problem is mathematically defined as follows. Minimize:

$$Min \left[K_T \sum_{i=1}^{n-1} h_i + K_J \sum_{j=1}^N \int_0^T (\ddot{q}_j(t))^2 dt \right] \tag{1}$$

Subject to:

$$\begin{cases} \dot{q}_j(t) \leq \dot{q}_j^{\max}, j = 1, 2, \dots, N \\ \ddot{q}_j(t) \leq \ddot{q}_j^{\max}, j = 1, 2, \dots, N \\ \ddot{q}_j(t) \leq \ddot{q}_j^{\max}, j = 1, 2, \dots, N \end{cases} \tag{2}$$

The meaning of the symbols mentioned above is explained in Table 1.

Noting that the two objectives have contradictory effects. Traveling time minimization would lead to faster but less smooth trajectories while reducing the jerk objective will lead to smoother but slower trajectories. This stated problem is solved with sequential quadratic programming (SQP) techniques available in MATLAB™. The trajectory solving of the above defined optimization problem must meet not only the continuity constraints appearing in Eq. (2) but also the interpolation conditions for all the knots, as well as the kinematic boundary conditions.

3 Proposed method

3.1 Definition of the trajectory by means of multiquadric radial basis functions

The interpolation functions are probably the most frequent operations used in many engineering and sciences problems such as the approximation of a function, a common idea implied by these functions to represent the approximate function as a linear combination of a set of points and a basis function which follows the interpolation condition [47].

A fundamental interpolation problem may be presented as follows:

Given a data set $(x_i, y_i), i = 1, \dots, n$ with x_i and y_i being coordinates of a point that assumed to be all different from each other, find a continuous function f such that:

$$f(x_i) = y_i, i = 1, \dots, n \tag{3}$$

Indeed, to solve this interpolation problem, Hardy and Rolland [48, 49] introduced an original idea based on the MQ-RBF’s method for the function approximation, which were proved to be the best interpolation functions in more than thirty functions [50]. An RBF is a real-valued function whose value depends only on the radial distance from the origin of any specific reference point called as “center,” which is known as radial basis function (RBF) and has the form:

$$\phi(x, c) = \phi(\|x - c\|) \tag{4}$$

where ϕ is the radial basis function, $\| \cdot \|$ denotes the Euclidean distance, x coordinate of a point, and c is the RBF center. Any multivariate functions which can be expressed as univariate functions of the Euclidean norm $\| \cdot \|$ are called radial basis functions.

Table 1 Meaning of symbols

Symbol	Description
JM	Jerk index objective function
N	Number of robot joints
T	Traveling time objective function
n	Number of interpolation points
t	Independent time variable
h_i	The time interval between two via-points
K_T	Weight of the term proportional to the execution time
K_J	Weight of the term proportional to the jerk
$\{\dot{q}_j(t), \ddot{q}_j(t), \ddot{q}_j(t)\}$	Velocity, acceleration, and jerk of the jth joint
$\{\dot{q}_j^{max}, \ddot{q}_j^{max}, \ddot{q}_j^{max}\}$	Velocity, acceleration, and jerk limits for joint j

The RBF basis function could be piecewise infinitely smooth and may have the shape parameter σ in multiquadric and Gaussian cases, or smooth and may not have it, for example, as in polyharmonic splines. The shape parameter σ , used to tune the overall shape of the RBF basis function in the approximation process, is a free positive constant needed to be specified by the user in advance [51].

In this study, we are especially interested in the classical MQ-RBF, which is given by [52]:

$$\phi(r) = \sqrt{r^2 + \sigma^2} \tag{5}$$

with:

$$r = \|x - c\|$$

and

$$\sigma > 0$$

where the 1st, second, and third derivatives are expressed as follows:

$$\frac{\partial \phi(r)}{\partial x} = \frac{(x - c)}{\sqrt{r^2 + \sigma^2}} = \frac{(x - c)}{\phi(r)} \tag{6}$$

$$\frac{\partial^2 \phi(r)}{\partial x^2} = \frac{1}{\sqrt{r^2 + \sigma^2}} - \frac{(x - c)^2}{(r^2 + \sigma^2)^{3/2}} = \left(1 - \frac{(x - c)^2}{\phi(r)^2}\right) \frac{1}{\phi(r)} \tag{7}$$

$$\frac{\partial^3 \phi(r)}{\partial x^3} = \frac{3(x - c)^3}{(r^2 + \sigma^2)^{5/2}} - \frac{3(x - c)}{(r^2 + \sigma^2)^{3/2}} = \left(\frac{3(x - c)^3}{\phi(r)^2} - \frac{3(x - c)}{1}\right) \frac{1}{\phi(r)^3} \tag{8}$$

Hence, to approximate the interpolating function f everywhere by a linear combination of certain RBF ϕ_j , gives, [52]

$$f(x) = \sum_{j=1}^n \omega_j \phi_j(\|x - c_j\|) \tag{9}$$

where $f(x)$ is a sum of n RBF's, each associated with a center $c_j = x_j$ and weighted by an appropriate weight ω_j . x is the input which we wish to evaluate our function f . By choosing interpolate nodes, we can approximate the function $f(x)$:

$$y_i = \sum_{j=1}^n \omega_j \phi_j(\|x_i - x_j\|), \quad i = 1, \dots, n \tag{10}$$

After some algebra, solving the interpolation problem leads to a system of linear equations is obtained as follows form:

$$KW = Y, \tag{11}$$

with

$$\begin{cases} W = [\omega_1, \dots, \omega_n]^T \\ Y = [y_1, \dots, y_n]^T \end{cases}$$

Note that the entries of the interpolation matrix are given by $A_{ij} = \phi_j(x_i); i, j = 1, \dots, n$ and constitute a positive definite matrix. Thus, using relation Eq. (5), we can define the coefficient matrix as follows:

$$K = \begin{bmatrix} 1 & \phi_2(x_1) & \dots & \phi_n(x_1) \\ \phi_1(x_2) & 1 & \dots & \phi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_n) & \phi_2(x_n) & \dots & 1 \end{bmatrix} \tag{12}$$

3.2 Boundary conditions

In robot trajectory context, the time scale is referred by the independent variable x and the n points requirement to be interpolated in increasing order, i.e., $x_1 < x_2 < \dots < x_m$. Likewise, we are generally interested in assigning the velocity and acceleration for the start and end knots of trajectory, i.e.,

$$v_1 = \frac{\partial f(x = x_1)}{\partial x}, \quad v_n = \frac{\partial f(x = x_n)}{\partial x}, \tag{13}$$

$$a_1 = \frac{\partial^2 f(x = x_1)}{\partial x^2}, \quad a_n = \frac{\partial^2 f(x = x_n)}{\partial x^2} \tag{14}$$

where velocities $\{v_1, v_n\}$ and accelerations $\{a_1, a_n\}$ are given data of the problem.

To accomplish this goal two extra knots (virtual via-points) with the associated RBFs should be introduced near each extreme point of RBF's so that the boundary conditions for velocity and acceleration can be respected. Therefore, the relation Eq. (9) rewrite as follows:

$$f(x) = \sum_{j=1}^{n+4} \omega_j \phi_j(\|x - c_j\|) \tag{15}$$

The new elements defining the associated linear system are as follows:

- The coefficients matrix $K [n + 4 \times n + 4]$:

$$K = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \tag{16}$$

with:

$$k_1 = \begin{bmatrix} 1 & \phi_2(x_1) & \dots & \phi_n(x_1) \\ \phi_1(x_2) & 1 & \dots & \phi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_n) & \phi_2(x_n) & \dots & 1 \\ \phi'_1(x_{n+1}) & \phi'_2(x_{n+1}) & \dots & \phi'_n(x_{n+1}) \\ \phi'_1(x_{n+2}) & \phi'_2(x_{n+2}) & \dots & \phi'_n(x_{n+2}) \\ \phi''_1(x_{n+3}) & \phi''_2(x_{n+3}) & \dots & \phi''_n(x_{n+3}) \\ \phi''_1(x_{n+4}) & \phi''_2(x_{n+4}) & \dots & \phi''_n(x_{n+4}) \end{bmatrix} \tag{17}$$

$$k_2 = \begin{bmatrix} \phi_{n+1}(x_1) & \phi_{n+2}(x_1) & \phi_{n+3}(x_1) & \phi_{n+4}(x_1) \\ \phi_{n+1}(x_2) & \phi_{n+2}(x_2) & \phi_{n+3}(x_2) & \phi_{n+4}(x_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{n+1}(x_n) & \phi_{n+2}(x_n) & \phi_{n+3}(x_n) & \phi_{n+4}(x_n) \\ \phi'_{n+1}(x_{n+1}) & \phi'_{n+2}(x_{n+1}) & \phi'_{n+3}(x_{n+1}) & \phi'_{n+4}(x_{n+1}) \\ \phi'_{n+1}(x_{n+2}) & \phi'_{n+2}(x_{n+2}) & \phi'_{n+3}(x_{n+2}) & \phi'_{n+4}(x_{n+2}) \\ \phi''_{n+1}(x_{n+3}) & \phi''_{n+2}(x_{n+3}) & \phi''_{n+3}(x_{n+3}) & \phi''_{n+4}(x_{n+3}) \\ \phi''_{n+1}(x_{n+4}) & \phi''_{n+2}(x_{n+4}) & \phi''_{n+3}(x_{n+4}) & \phi''_{n+4}(x_{n+4}) \end{bmatrix} \tag{18}$$

Noting that the previous four rows of matrix K are calculated via relations given in Eqs. (6) and (7). Moreover, the values of the four extra knots added to RBF’s trajectory are given by:

$$\begin{aligned} x_{n+1} &= x_1 + 0.02(x_2 - x_1); & x_{n+2} &= x_n - 0.02(x_n - x_{n-1}); \\ x_{n+3} &= x_1 + 0.04(x_2 - x_1); & x_{n+4} &= x_n - 0.04(x_n - x_{n-1}) \end{aligned} \tag{19}$$

- The unknown vector of the linear system:

$$W[n + 4 \times 1] = [\omega_1, \dots, \omega_n, \omega_{n+1}, \omega_{n+2}, \omega_{n+3}, \omega_{n+4}]^T \tag{20}$$

and where the vector Y becomes:

$$Y[n + 4 \times 1] = [y_1, \dots, y_n, v_1, v_n, a_1, a_n]^T \tag{21}$$

3.3 Optimization process consideration

In the optimization problem, the solution to this problem was denoted as the vector of optimization variables $F = (h_1, \dots, h_{n-1}, \sigma_1, \dots, \sigma_N)$. That is, the issue is to find a set of optimum values for the time intervals h_j ($h_j = t_{j+1} - t_j, j = 1, \dots, n - 1$), and the suitable shape parameters σ_i ($i = 1, \dots, N$) such as the transfer time is minimized between each pair of adjacent knots. Kinematic constraints on velocity, acceleration, and jerk for all joints

are respected using the MQ-RBF interpolation technique. Also, it remains to describe the expression of the objective function Eq. (1) in order, to be able to implement the algorithm for the trajectory optimization constructed with MQ-RBFs. So, the objective function Eq. (1) can be defined in terms of MQ-RBF and formulated as.

$$F_{obj} = K_T T + K_J JM \tag{22}$$

where the execution time

$$T = \sum_{i=1}^{n-1} h_i \tag{23}$$

and absolute mean jerk

$$JM = \sum_{j=1}^N \sqrt{\frac{1}{T} \int_0^T \ddot{q}_j^2(t) dt} \tag{24}$$

Therefore, to be capable to compare the results obtained by our method, the values of the weighting coefficients K_T and K_J presented in the expression of the objective function Eq. (22) have been adjusted by choosing $K_J = 0$ permits to find a minimum-time trajectory Eq. (23), while setting $K_T = 0$ permits to find a minimum-jerk trajectory profile is found Eq. (24).

4 Comparative examples

4.1 Trajectory for pick-and-place tasks

As previously stated, the objective here is to find a trajectory that connects a beginning and final configuration while also satisfying other specified constraints at the endpoints (null limit velocity and acceleration). To achieve this goal, we will use two knots associated with the limit configurations, and four extra-knots are inserted according Eq. (19). The trajectory found by MQ-RBF is then compared to those obtained by using [52]:

- Gaussian RBF (Ga-RBF):

$$\varphi(r) = \exp\left(-\frac{1}{2} \frac{r^2}{\sigma^2}\right) \tag{25}$$

- Generalized inverse multiquadric RBF (GIMQ-RBF):

$$\varphi(r) = \frac{1}{(r^2 + \sigma^2)^2} \tag{26}$$

- Inverse quadric RBF (IQ-RBF):

$$\varphi(r) = \frac{1}{\sqrt{r^2 + \sigma^2}} \tag{27}$$

- Quintic RBF (Qnt-RBF):

$$\varphi(r) = r^5 \tag{28}$$

As a first illustrative example, a single link trajectory is considered it has to connect the limit configurations $q_i = 0$, $q_f = \pi$ and the employed time interval $t_f = 1.5$ s. Also, the shape parameter σ is taken the same for all applied radial basis functions and we suppose $\sigma = t_f$. Trajectory plots of the five RBF's which connect the endpoints are given in Fig. 1. The corresponding profiles of velocity, acceleration, and jerk are also plotted in Fig. 1, for purposes of comparison. Table 2 reports the extreme recorded values.

Note from Fig. 1 that the position profiles for the five RBF's models are not significantly different and they link the

Table 2 Maximum values of velocity, acceleration, and jerk recorded during pick-and-place tasks. (The bold values indicate the lowest values)

Radial basis functions	V_{max} (deg/s)	A_{max} (deg/s ²)	J_{max} (deg/s ³)
Ga-RBF	4.0203	8.2790	48.4002
GIMQ-RBF	4.4951	9.9652	47.7309
IQ-RBF	4.4207	9.6465	45.2121
MQ-RBF	4.3015	9.1527	40.9374
Qnt-RBF	3.9270	8.0607	55.8505

initial and final positions in a smooth way, with no overshoot beyond the endpoints. From velocity, acceleration, and jerk points of view, it is clear that Ga-RBF trajectory behaves like the Qnt-RBF model and that Qnt-RBF trajectory has a better

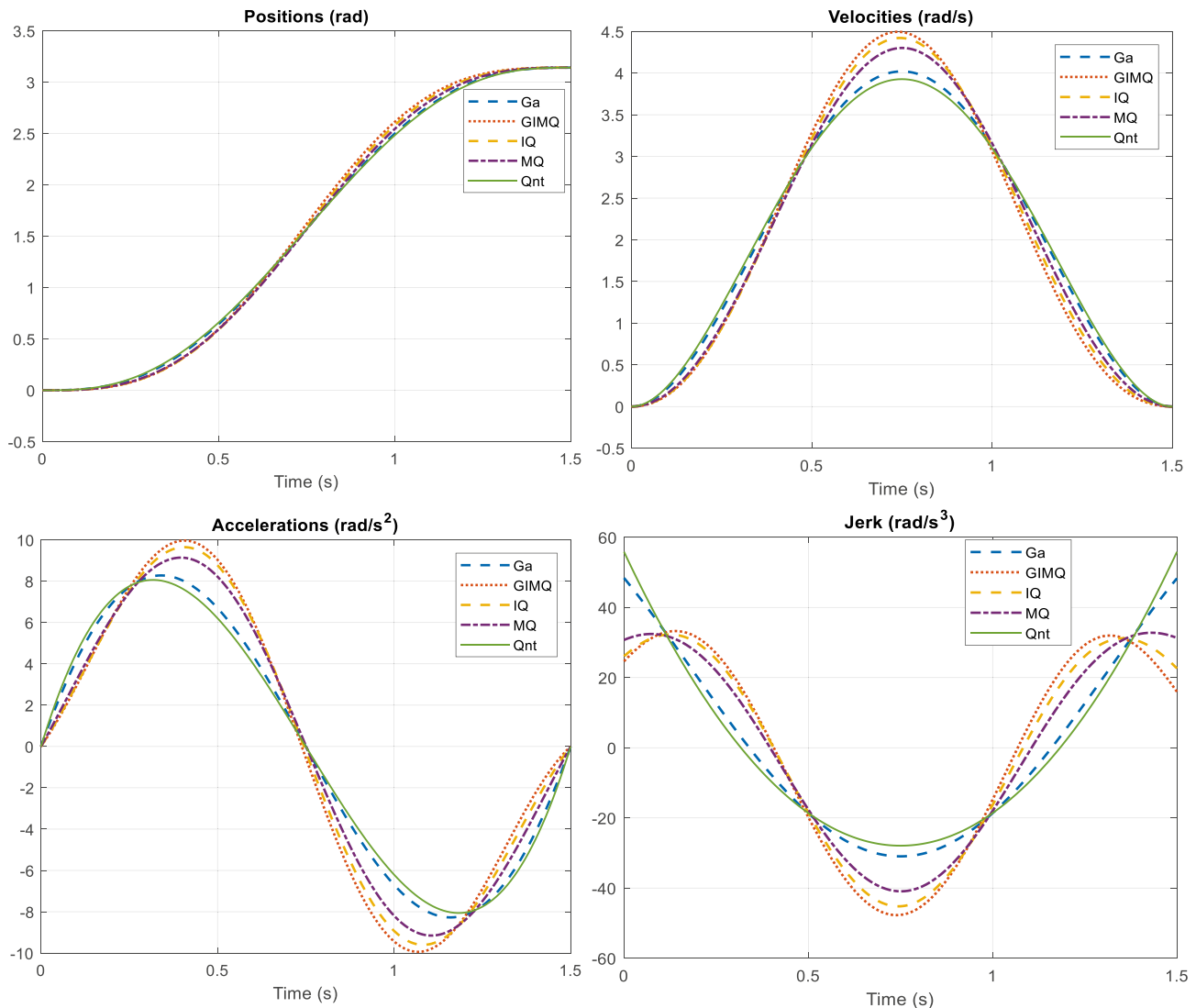


Fig. 1 Trajectory model for pick-and-place tasks

velocity and acceleration profiles. However, this superiority is paid by having significantly higher values in the jerk function. It can be noted that MQ-RBF trajectory is close to those of the GIMQ-RBF and IQ-RBF models but with better scores in terms of extreme values of jerk compared with other RBF's trajectories, as indicated in Table 2.

4.2 Trajectory for continuous tasks

As a second illustrative example, a single link trajectory is considered for planning trajectory in continuous tasks as

in Ref. [18]. The trajectory has to connect the following angles (in degrees) $q = [120, 60, 80, 120, 0]$. The employed time intervals are 2 s between each pair of adjacent knots. The objective of this example is to test the use of MQ-RBF for planning joint trajectories and to compare the results with those obtained by using:

- Gaussian RBF (Ga-RBF): implemented according to Ref. [43], and
- Cubic spline (CS): implemented according to Ref. [53]

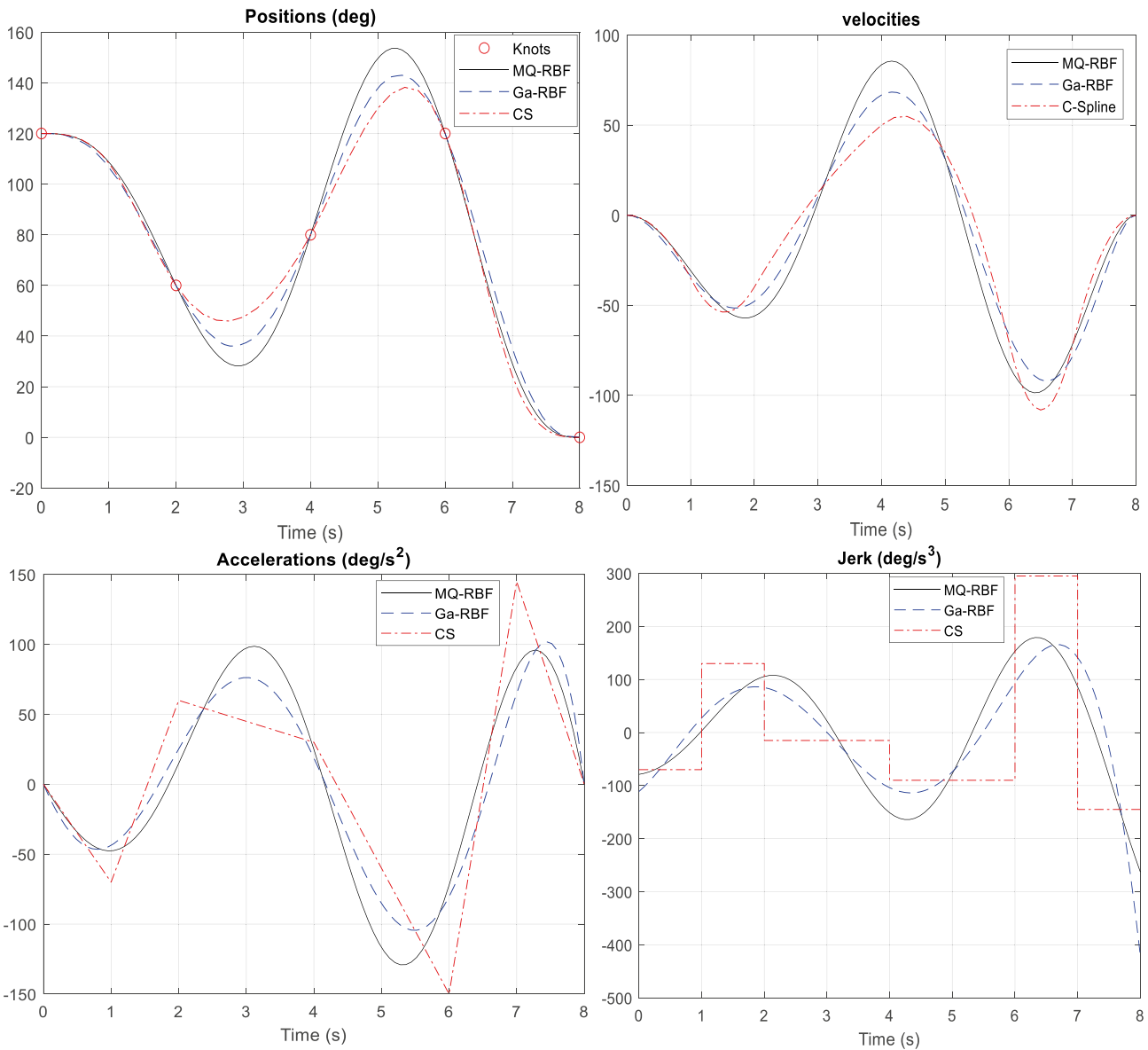


Fig. 2 Trajectory model for continuous tasks

Table 3 Maximum values of velocity, acceleration, and jerk recorded during continuous tasks. (The bold values indicate the lowest values)

Trajectory model	V_{max} (deg/s)	A_{max} (deg/s ²)	J_{max} (deg/s ³)
MQ-RBF	98.9206	130.3330	258.2874
Ga-RBF	92.0799	104.5160	420.4119
CS	108.1250	150.0000	295.0000

Note that the only drawback of the cubic splines is just the presence of the discontinuity in the acceleration function at the extreme points. This can be avoided, by adding two free extra-knots in the second and penultimate position. Trajectory plots of the three interpolation functions (MQ-RBF, Ga-RBF, and CS) which pass through the five sets of knots with and their derivative are given in Fig. 2. Table 3 summarizes the extreme recorded values of velocity, acceleration, and jerk.

Indeed, from Fig. 2 the position and velocity profiles of the MQ-RBF and Ga-RBF trajectories behave like the cubic spline. Also, it can be noted in Fig. 2 that Ga-RBF trajectory determined according the procedure described in [43] has a better velocity and acceleration profile. However, this superiority is paid by having an oscillating jerk profile with significantly higher values, compared with the MQ-RBF trajectory which has the better scores in terms of extreme values of jerk, as indicated in Table 3.

5 Generating trajectories results and interpretations

5.1 Example 1: Minimum time trajectory

The objective function that contains the minimum-time Eq. (23) for the PUMA 560 robot is considered in this

Table 5 Kinematic constraints of the robot joints

Constraints	Joint					
	1	2	3	4	5	6
Velocity (°/s)	100	95	100	150	130	110
Acceleration (°/s ²)	45	40	75	70	90	80
Jerk (°/s ³)	60	60	55	70	75	70

example. The goal is to plan optimal trajectories and compare the obtained results with those given in Chettibi's recent works [43]. The knot angle values in the joint space are reported in Table 4, and the kinematic limits of the joints are given in Table 5. Note that the velocity and acceleration at start knots and end knots were set as zero.

The obtained total traveling time of the trajectory is $T = 17.2347$ s, the optimum values of time intervals are $h_1 = 3.3201$ s, $h_2 = 1.1645$ s, $h_3 = 1.9295$ s, $h_4 = 2.3207$ s, $h_5 = 2.4683$ s, $h_6 = 2.9501$ s, and $h_7 = 3.0816$ s and the optimum values MQ-RBF shape parameters are $\sigma_1 = 3.0214$, $\sigma_2 = 9.0955$, $\sigma_3 = 2.4557$, $\sigma_4 = 1.5580$, $\sigma_5 = 8.5448$, and $\sigma_6 = 2.9987$. The joint trajectories and their derivatives of the 6-DOF robot manipulator in minimum time which pass through the six via-points using the MQ-RBF interpolation, are given in Fig. 3.

The traveling time found by the Gaussian RBF interpolation in [43] is about 17.7107 s. In Refs. [30, 54], it is stated a minimum time of about 18.451 s using cubic spline and 21.964 s using cubic B-spline, respectively. The results obtained by the MQ-RBF interpolation technique are comparable with those from Chettibi [43], taking into account end conditions and kinematic constraints. This first example illustrates that the proposed technique is capable of achieving shorter production times. This appears to be an essential property that is highly desirable in the industrial field.

Table 4 Joint space knots for trajectory planning

Knot	Joint (degrees)					
	1	2	3	4	5	6
1	10	15	45	5	10	6
2	60	25	180	20	30	40
3	75	30	200	60	-40	80
4	130	-45	120	110	-60	70
5	110	-55	15	20	10	-10
6	100	-70	-10	60	50	10
7	-10	-10	100	-100	-40	30
8	-50	10	50	-30	10	20

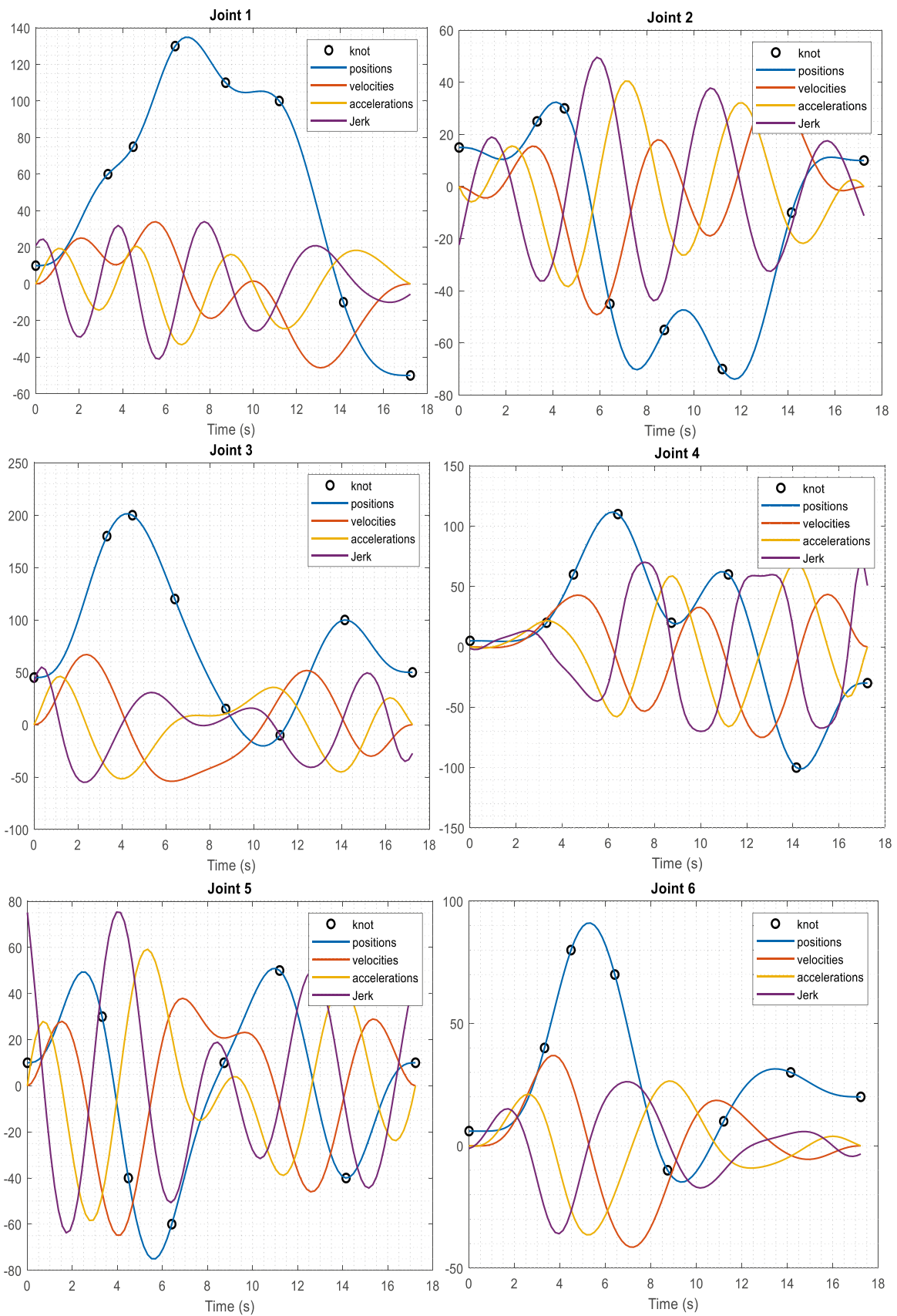


Fig. 3 Optimized trajectory with minimum time

Table 6 Joint space knots for trajectory planning

Knot	Joint (degrees)					
	1	2	3	4	5	6
1	−10	20	15	150	30	120
2	60	50	100	100	110	60
3	20	120	−10	40	90	100
4	55	35	30	10	70	25

5.2 Example 2: Minimum time-jerk trajectory

The second example aims to plan a smooth trajectory planning for the industrial robot PUMA 560 with six revolute joints and to compare the obtained results with the algorithms given by two important earlier studies [14, 34]. For each knot point, there are four different knot angles ($n=4$), although the authors [14, 34] utilized joint space values reported in Table 6 while respecting kinematic constraints given in Table 7. Note that the velocity and acceleration were set as zero at the beginning and end points of the trajectory.

In order to be able to compare the results yielded by our method, the values of the weighting coefficients K_T and K_J presented in the expression of the objective function Eq. (22) have been adjusted as Gasparetto and Zanotto [34], so that the traveling time of the algorithm described in this example and the technique presented in [34] would be the same (about 9.1 s). The obtained total traveling time of the trajectory is $T=9.0975$ s, the optimum values of time intervals are $h_1=3.3393$ s, $h_2=2.5904$ s, and $h_3=3.3393$ s, and the optimum values of MQ-RBF shape parameters are $\sigma_1=5.5483$, $\sigma_2=3.3848$, $\sigma_3=5.3837$, $\sigma_4=6.0000$, $\sigma_5=5.9905$, and $\sigma_6=5.4499$.

The 6-DOF robot manipulator's trajectories and their derivatives up to the third-order are presented in Fig. 4.

These figures show that the two first derivatives of joint trajectories curves are null in the beginning and end-point of the trajectory. They show that all kinematic performances are met. Figures 5, 6, 7, and 8 show the mean kinematic values (velocity, acceleration, and jerk) for all joints and their average values, respectively, compared with those obtained by the method proposed by [14, 34].

Table 8 illustrates the maximum joint velocities, accelerations, and jerks obtained from our method. It can be noticed from these values that all kinematic performances for the robot manipulator are met.

Table 9 illustrates the mean kinematic values of velocity, acceleration, and jerk and their average values for all joints, compared with those results from Gasparetto and Piazzzi works [14, 34]. Gasparetto and Zanotto [34] used the cubic splines interpolation with the SQP method to minimize the time-jerk objective function whereas in the front of Piazzzi's work [14] which the trajectory interpolated by means of minimum jerk (MJ) cubic splines. In this study, as shown in Table 9, the mean jerks of joint 1, 2, 3, 5, and 6 yielded by MQ-RBF are lower than those yielded by Gasparetto's technique with 14%, 7%, 15%, 1%, and 11% respectively, and the results yielded by Piazzzi's method are lower with 16%, 11%, 16%, 1%, and 16% respectively. It can be easily said that the MQ-RBF technique performed better than those presented in the earlier studies [14, 34] in the aspect of jerk value.

Table 7 Kinematic constraints of the robot joints

Constraints	Joint					
	1	2	3	4	5	6
Velocity ($^{\circ}/s$)	100	95	100	150	130	110
Acceleration ($^{\circ}/s^2$)	60	60	75	70	90	80
Jerk ($^{\circ}/s^3$)	60	66	85	70	75	70

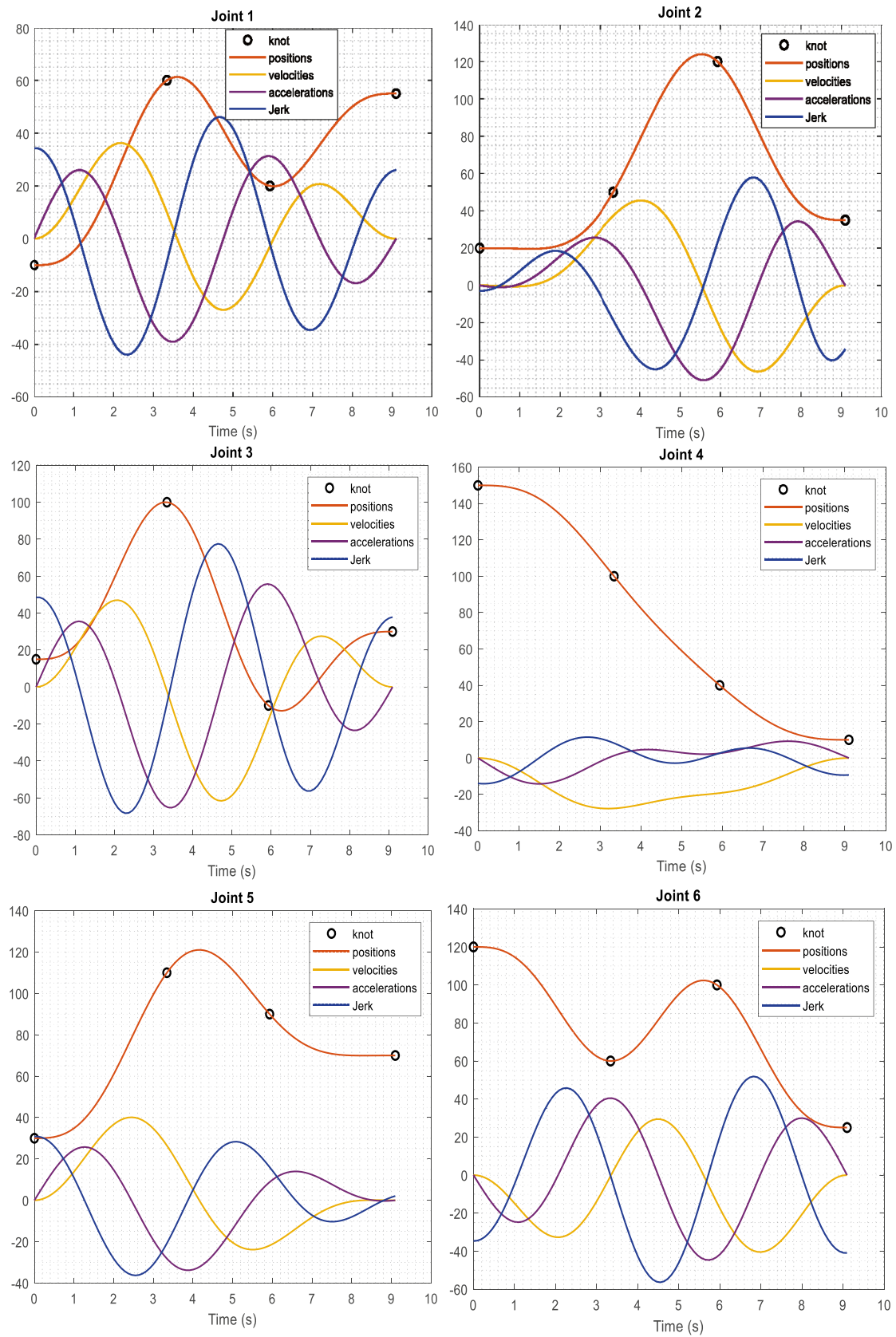


Fig. 4 Optimized trajectory with minimum time-jerk

Fig. 5 Mean velocity values

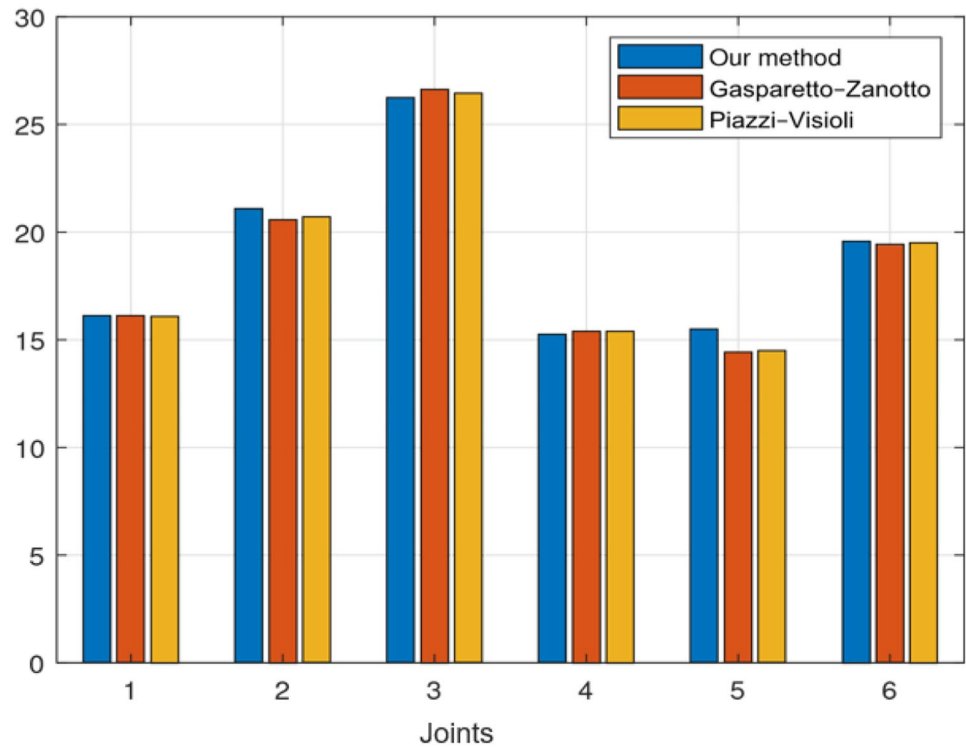


Fig. 6 Mean acceleration values

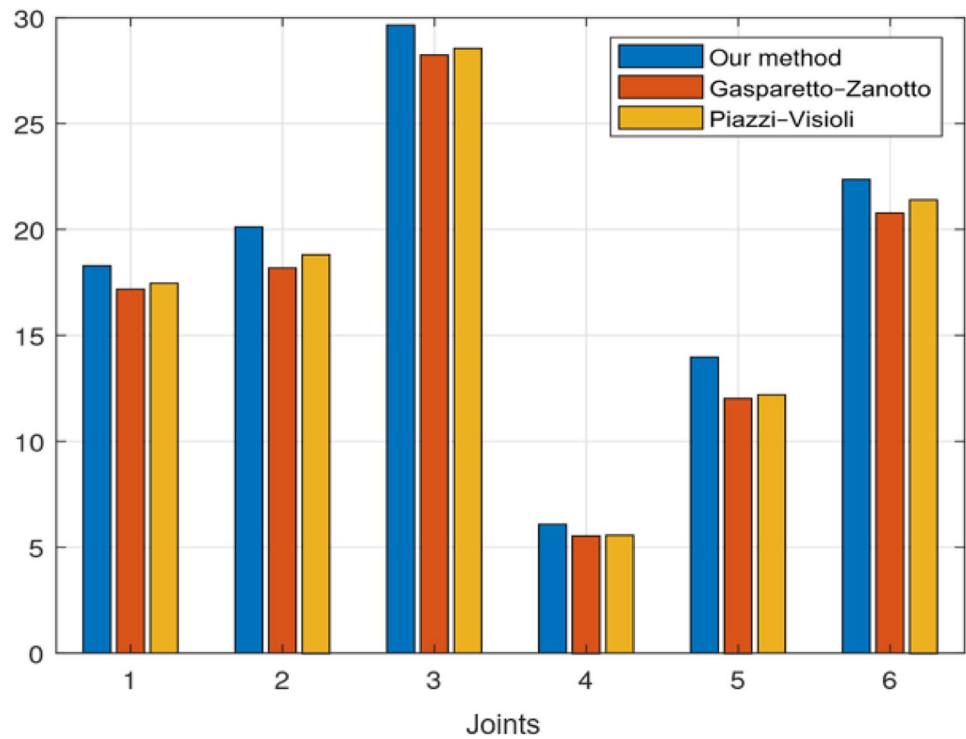


Fig. 7 Mean jerk values

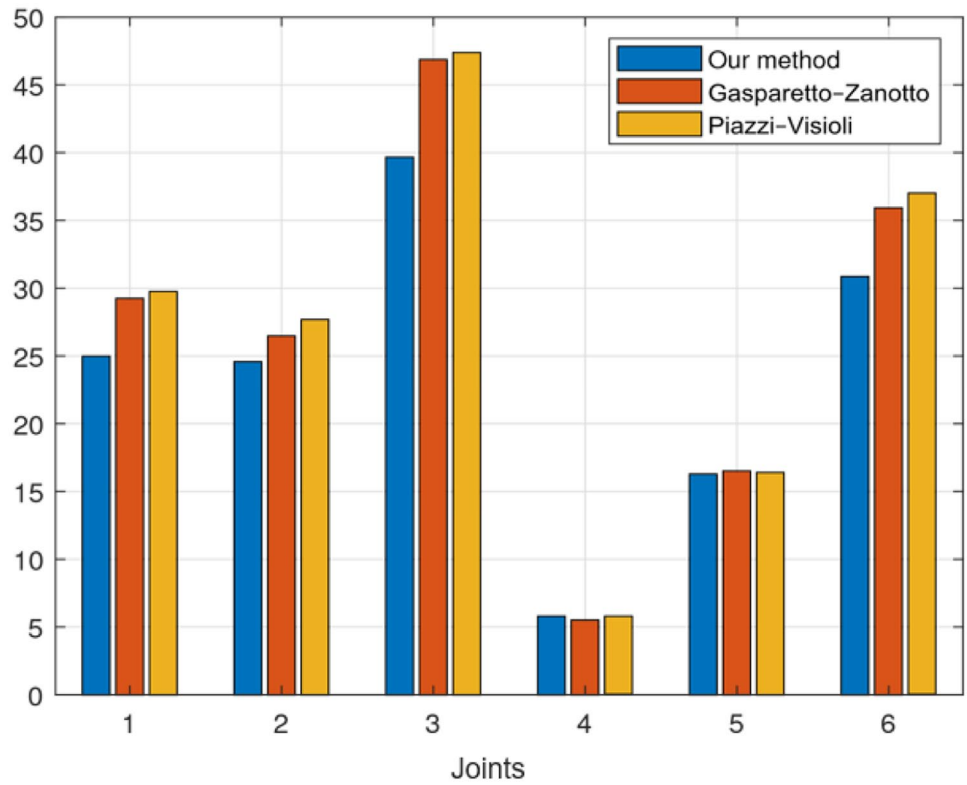


Fig. 8 Average kinematic values

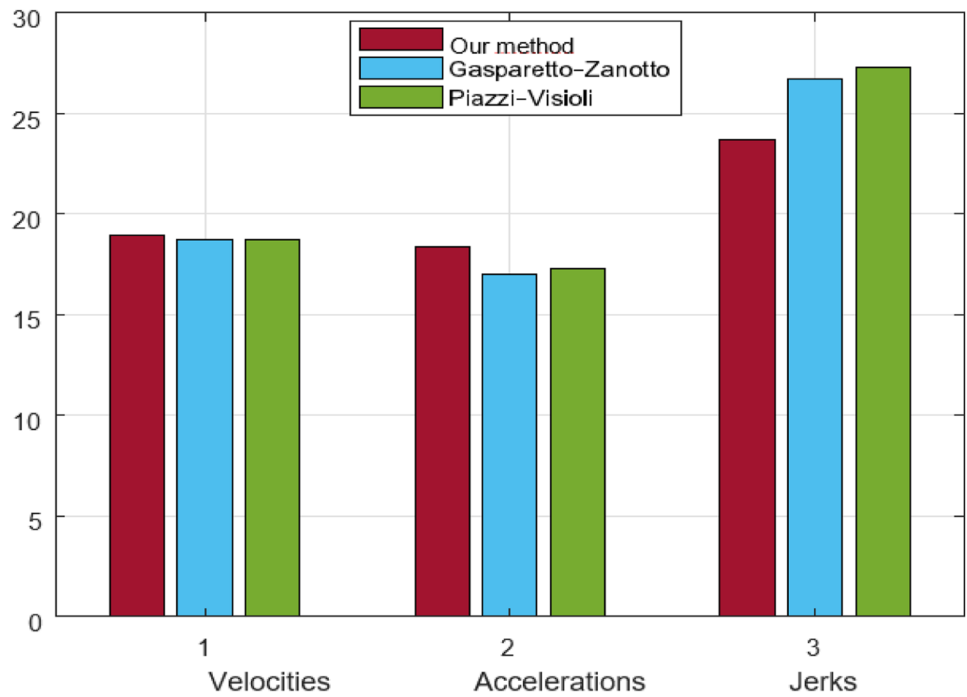


Table 8 Maximum kinematic values

Algorithm		Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Our method	V_{max} ($^{\circ}/s$)	36.27	46.22	61.60	27.82	40.12	40.48
	A_{max} ($^{\circ}/s^2$)	38.98	50.86	65.31	14.22	33.79	44.68
	J_{max} ($^{\circ}/s^3$)	46.11	57.93	77.49	14.13	36.30	56.40

Table 9 Mean kinematic values

Algorithm		Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Average
Our method	V_{mean} ($^{\circ}/s$)	16.09	21.07	26.21	15.23	15.47	19.56	18.94
	A_{mean} ($^{\circ}/s^2$)	18.27	20.10	29.62	6.05	13.94	22.34	18.39
	J_{mean} ($^{\circ}/s^3$)	24.95	24.55	39.63	5.38	16.24	30.82	23.59
Gasparetto–Zanotto	V_{mean} ($^{\circ}/s$)	16.10	20.57	26.61	15.38	14.40	19.40	18.74
	A_{mean} ($^{\circ}/s^2$)	17.15	18.15	28.23	5.53	12.03	20.76	16.97
	J_{mean} ($^{\circ}/s^3$)	29.24	26.45	46.85	5.48	16.48	35.90	26.73
Piazzi–Visioli	V_{mean} ($^{\circ}/s$)	16.08	20.69	26.42	15.38	14.46	19.50	18.75
	A_{mean} ($^{\circ}/s^2$)	17.45	18.80	28.52	5.55	12.18	21.39	17.31
	J_{mean} ($^{\circ}/s^3$)	29.73	27.68	47.35	5.75	16.38	36.97	27.31

6 Conclusion

A new technique, using multiquadric radial basis functions (MQ-RBFs), has been presented to construct robot trajectories. The trajectory planning is performed in the JS, and the transfers with via-points can be simply attained satisfying imposed boundary conditions. Considering this fact, the proposed approach was tested and verified in diverse cases and with others interpolation functions such as RBF's and splines. Also, the trajectory constructed with MQ-RBF is optimized; using SQP technique and a minimum-time, smooth trajectory profile is found for the 6-DOF PUMA 560 robot that meets kinematics constraints. The proposed methodology has been tested in simulation and the results are compared with important approaches proposed by the previous authors [14, 30, 34, 43, 54].

The performance of the proposed technique can be summarized as follows:

1. The trajectory construction using MQ-RBFs allows to generate smooth trajectories with satisfying imposed constraints on the initial and final velocities and accelerations.
2. The proposed approach is capable to construct joint trajectories for robot manipulators as good as cubic splines, cubic B-spline and Gaussian-RBF in terms of traveling time but with continuous derivatives and significantly decreased jerk.
3. The proposed technique also makes it possible to process a large number of via-points in a trajectory with simple implementation.

4. MQ-RBF technique provides minimum jerk index with the same transfer time compared to previous studies.

Further research will investigate and exploit the benefits of the proposed technique for planning the trajectory of parallel manipulators and CNC machines. Other innovative multi-objective optimization techniques will also be taken into account. Optimal trajectory planning can also be improved by using other optimization techniques. In addition, other Hybrid RBF (HRBF) interpolation methods such as the Hybrid Gaussian–cubic radial basis function can be considered to obtain smooth trajectories.

Author contribution Bendali Nadir: performed the algorithm design, code writing, and was a major contributor in writing the manuscript. All authors read and approved the final manuscript.

Funding This work was supported by “La Direction Générale de la Recherche Scientifique et du Développement Technologique (DGRSDT)” of Algeria to whom we are grateful.

Availability of data and material All data generated or analyzed during this study are included in this published article.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Competing interests The authors declare no competing interests.

References

- Angeles J (2014) Fundamentals of robotic mechanical systems theory methods and algorithms. Springer International Publishing
- Gasparetto A, Boscaroli P, Lanzutti A, Vidoni R (2015) Path planning and trajectory planning algorithms: a general overview. Motion and operation planning of robotic systems. Springer, pp 3–27
- Kahn ME, Roth B (1971) The near-minimum-time control of open-loop articulated kinematic chains. *J Dyn Syst Meas Control Trans ASME* 164–172
- Kim BK, Shin KG (1985) Minimum-time path planning for robot arms and their dynamics. *IEEE Trans Syst Man Cybern* 213–223
- Bobrow JE, Dubowsky S, Gibson JS (1985) Time-optimal control of robotic manipulators along specified paths. *Int J Rob Res* 4(3):3–17
- Shin KG, McKay ND (1985) Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Trans Auto Cont*
- Constantinescu D, Croft EA (2000) Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *J Robot Syst*
- Xiao Y, Du Z, Dong W (2012) Smooth and near time-optimal trajectory planning of industrial robots for online applications. *Ind Rob*
- Bendali N, Ouali M (2018) Optimization the trajectories of robot manipulators along specified task. *Int J Intell Eng Sys*
- Bendali N, Ouali M, Gellal K, (2015) Optimal trajectory planning under kino-dynamics constraints for a 6-DOF PUMA 560. In: Proceedings of the The International Conference on Engineering, Turkey
- Abu-Dakka FJ, Valero FJ, Suñer JL, Mata V (2015) A direct approach to solving trajectory planning problems using genetic algorithms with dynamics considerations in complex environments. *Robotica* 33(3):669–683
- Bendali N, Ouali M, Kifouche A (2014) A new method for time-jerk optimal trajectory planning under kino-dynamic constraint of robot manipulators in pick-and-place operations. *IAES Int J Robot Autom*
- Kyriakopoulos KJ, Saridis GN (1988) Minimum jerk path generation
- Piazzi A, Visioli A (2000) Global minimum-jerk trajectory planning of robot manipulators. *IEEE Tran Indu Elec*
- Piazzi A, Visioli A (1997) Interval algorithm for minimum-jerk trajectory planning of robot manipulators. In: Proceedings of the IEEE Conference on Decision and Control
- Liny HI, Liu YC (2011) Minimum-jerk robot joint trajectory using particle swarm optimization. In: Proceedings - 1st International Conference on Robot, Vision and Signal Processing, RVSP
- Lin HI (2014) A fast and unified method to find a minimum-jerk robot joint trajectory using particle swarm optimization. *J Intell Robot Syst: Theo Appl*
- Simon D, Isik C (1993) A trigonometric trajectory generator for robotic arms. *Int J Control*
- Simon D (2004) Neural networks for optimal robot trajectory planning. In: Handbook of Neural Computation
- Simon D (1993) The application of neural networks to optimal robot trajectory planning. *Rob Auton Syst*
- Lu S, Zhao J, Jiang L, Liu H (2017) Solving the time-jerk optimal trajectory planning problem of a robot using augmented lagrange constrained particle swarm optimization. *Math Probl Eng*
- Lu S, Zhao J, Jiang L, Liu H (2017) Time-jerk optimal trajectory planning of a 7-DOF redundant robot. *Turk J Electric Eng Comput Sci*
- Jiang L, Lu S, Gu Y, Zhao J (2017) Time-jerk optimal trajectory planning for a 7-DOF redundant robot using the sequential quadratic programming method. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
- Rout A, Dileep M, Mohanta GB, Deepak BBVL, Biswal BB (2018) Optimal time-jerk trajectory planning of 6 axis welding robot using TLBO method. *Procedia Comput Sci* 133:537–544
- Cong M, Xu X, Xu P (2010) Time-jerk synthetic optimal trajectory planning of robot based on fuzzy genetic algorithm. *I J Int Syst Tech Appl*
- Gasparetto A, Lanzutti A, Vidoni R, Zanutto V (2012) Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning. *Robot Comput Integr Manuf* 28(2):164–181
- Gasparetto A, Zanutto V (2007) A new method for smooth trajectory planning of robot manipulators. *Mech Mach Theory*
- Lombai F, Szederkenyi G (2008) Trajectory tracking control of a 6-degree-of-freedom robot arm using nonlinear optimization. In: International Workshop on Advanced Motion Control AMC
- Lombai F, Szederkenyi G (2009) Throwing motion generation using nonlinear optimization on a 6-degree-of-freedom robot manipulator. In: IEEE 2009 International Conference on Mechatronics ICM
- Lin CS, Chang PR, Luh JYS (1983) Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Trans Automat Contr*
- Thompson SE, Patel RV (1987) Formulation of joint trajectories for industrial robots using B-splines. *IEEE Trans Ind Electron*
- Liu H, Lai X, Wu W (2013) Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. *Robot Comput Integr Manuf*
- Ramabalan S, Saravanan R, Balamurugan C (2009) Multi-objective dynamic optimal trajectory planning of robot manipulators in the presence of obstacles. *Int J Adv Manuf Technol*
- Gasparetto A, Zanutto V (2008) A technique for time-jerk optimal planning of robot trajectories. *Robot Comput Integr Manuf*
- Huang J, Hu P, Wu K, Zeng M (2018) Optimal time-jerk trajectory planning for industrial robots. *Mech Mach Theory*
- Xu Z, Wei S, Wang N, Zhang X (2014) Trajectory planning with Bezier curve in Cartesian space for industrial gluing robot. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
- Fang S, Ma X, Zhao Y, Zhang Q, Li Y (2019) Trajectory planning for Seven-DOF robotic arm based on quintic polynomial. In: Proceedings - 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2019
- Xie Z, Wu P, Ren P (2016) A comparative study on the pick-and-place trajectories for a delta robot. In: Proceedings of the ASME Design Engineering Technical Conference
- Petrinec K, Kovacic Z (2007) Trajectory planning algorithm based on the continuity of jerk. In: Mediterranean Conference on Control and Automation
- Kucuk S (2017) Optimal trajectory generation algorithm for serial and parallel manipulators. *Robot Comput Integr Manuf*
- Liu H, Lai X, Wu W (2013) Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. *Robot Comput Integr Manuf* 29(2):309–317
- Buhmann MD, Levesley J (2004) Radial basis functions: theory and implementations. *Math Comput*
- Chettibi T (2019) Smooth point-to-point trajectory planning for robot manipulators by using radial basis functions. *Robotica*
- Mirinejad H, Inanc T (2016) An RBF collocation method for solving optimal control problems. *Rob Auton Syst*
- Mirinejad H, Inanc T, Zurada JM (2021) Radial basis function interpolation and Galerkin projection for direct trajectory optimization and costate estimation. *IEEE/CAA J Autom Sinica* 8(8):1380–1388

46. Zlajpah L, Petric T (2020) Generation of smooth Cartesian paths using radial basis functions. In: In International Conference on Robotics in Alpe-Adria Danube Region
47. Mishra PK, Nath SK, Sen MK, Fasshauer GE (2018) Hybrid Gaussian cubic radial basis functions for scattered data interpolation. *Comput Geosci*
48. Hardy RL (1971) Multiquadric equations of topography and other irregular surfaces. *J Geophys Res*
49. Hardy RL (1990) Theory and applications of the multiquadric-biharmonic method. *Comput Math App*
50. Franke R (1982) Scattered data interpolation: Tests of some methods. *Math Comput* 38(157):181–200
51. Mirinejad H, Inanc T (2015) A radial basis function method for direct trajectory optimization. In: Proceedings of the American Control Conference
52. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007) *Numerical recipes 3rd Edition: the Art of Scientific Computing*
53. Abu-Dakka FJ, Assad IF, Alkhdour RM, Abderahim M (2017) Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots. *Int J Adv Manuf Technol* 89(1–4):389–406
54. Wang CH, Horng JG (1990) Constrained minimum-time path planning for robot manipulators via virtual knots of the cubic B-spline functions. *IEEE Trans Automat Contr*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.