



An approach based on multiquadric radial basis functions for smooth trajectory planning of robotic manipulators with kinematic constraints

Nadir Bendali · Lorenzo Scalera ·
Arunachalam Sundaram · Abderrezak Said ·
Alessandro Gasparetto

Received: 8 February 2025 / Accepted: 12 April 2025
© The Author(s) 2025

Abstract This paper introduces a novel approach for planning smooth trajectories of robotic manipulators by leveraging multiquadric radial basis functions (MQ-RBFs). The proposed approach aims to achieve optimal trajectories by minimizing a multi-objective function that accounts for both time and jerk optimization. The MQ-RBF interpolation technique ensures that the trajectory meets velocity, acceleration, and jerk limits, while ensuring jerk continuity. Comparative evaluations are conducted in two cases: with and without optimization. In the first case, the MQ-RBF interpolation approach is compared with various RBF interpolation models. In the second case, the MQ-RBF trajectory

approach is compared with alternative state-of-the-art trajectory planning techniques, such as fifth-order B-splines and trigonometric spline functions, for generating optimal time-jerk trajectories for 6-joint robotic manipulators using optimization algorithms. Numerical and experimental results demonstrate the superior performance of the proposed technique in efficiently planning smooth trajectories compared to existing trajectory planning approaches and validate its effectiveness across various scenarios.

Keywords Interpolation techniques · MQ-RBF · Robotics · Time-jerk optimization · Trajectory planning

N. Bendali · A. Said
Faculty of Science and Technology, Khemis Miliana University,
Road of Theniet El Had, Khemis Miliana 44225, Algeria
e-mail: a.said@univ-dbkcm.dz

N. Bendali (✉)
Structural Mechanics Research Laboratory, Department of
Mechanical Engineering, University Blida-1, BP 270 Route
Soumâa, Blida 09000, Algeria
e-mail: n.bendali@univ-dbkcm.dz

L. Scalera (✉) · A. Gasparetto
Polytechnic Department of Engineering and Architecture, Uni-
versity of Udine, via delle Scienze 206, Udine 33100, Italy
e-mail: lorenzo.scalera@uniud.it

A. Gasparetto
e-mail: alessandro.gasparetto@uniud.it

A. Sundaram
School of Science, Engineering, and Environment, University of
Salford, SEE Building, Salford M5 4WT, UK
e-mail: a.sundaram@salford.ac.uk

1 Introduction

Trajectory planning is the foundation of motion control for robots to accomplish a desired task [1]. It is often approached from three perspectives: the temporal [2, 3], the motion smoothness [4], or addressing the trade-off between these two aspects [5]. Motion smoothness is a critical issue in precision manufacturing. Therefore, by maintaining a smooth trajectory, it is possible to prevent abrupt variations in load, prolong the lifespan of the joints, and position control precision can be enhanced [6]. To this end, many studies have been conducted on planning and optimizing trajectories by using different objective functions and constraints.

In robotics, trajectory planning is commonly described by polynomials [7–9], splines [10–29] or Bézier [30–33] interpolation functions, among other techniques. The employment of polynomial functions of higher order enables the generation of motion profiles characterized by enhanced smoothness. However, this approach comes with the drawback of a general rise in high speed values, assuming a constant total execution time. For instance, in [7], polynomial functions up to the 9th order are considered to present a description of the acceleration profile under the condition of an acceleration constraint. Mohamed et al. [8] employed 3rd, 4th, 5th, and 6th degree polynomials along with cycloidal and elliptical formulas to plan the optimal trajectory of a 3-DOF positioning micro-robot. An optimal time-jerk algorithm for trajectory planning subject to kinematic constraints is proposed by Bureerat et al. in [9], where a fifth-order polynomial function is adopted to connect two sub-paths, the beginning and intermediate positions, to the end position. However, only a few articles employed polynomials with degree greater than seven to describe displacement.

The cubic splines have been employed to construct the trajectory of robotic manipulators in many papers [10–14]. In their proposed approach, Gasparetto et al. [10, 11] applied cubic splines and the sequential quadratic programming (SQP) algorithm to optimize the bi-objective function related to the total execution time and the integral of the squared jerk along the whole trajectory. Similarly, in [4], the authors introduce a technique for optimizing minimum-jerk (MJ) trajectories to achieve a global solution using interval analysis. A series of cubic splines functions are used for interpolation. That approach can produce a continuous jerk, but it has very high computing requirements. The authors of [15, 16] provide a description of the trajectory planning to get minimal time along specified tasks using algebraic splines for industrial manipulators. Furthermore, Gallant and Gosselin [17] focused on the issue of increasing manipulator payload capacity for pick-and-place tasks in joint-space trajectory planning by optimizing cubic splines and Bernstein polynomials with a SQP optimization technique.

The applications of B-splines, just like cubic splines, have also been investigated in many research works with various algorithms, including cubic B-splines [18–21], quintic B-splines [5, 11, 22–27], and seventh-order B-splines [28, 29]. For instance, Saravanan et al. [18] developed a novel optimum trajectory plan-

ning method based on evolutionary theory using cubic B-splines. Zhang et al. [22] developed a trajectory planning approach for robots under obstacles using a multi-objective optimization technique to obtain an optimal time-jerk trajectory. Their approach uses fifth-order B-splines to construct joint trajectories with kinematic and obstacle constraints. Moreover, Huang et al. [27] proposed the NSGA-II optimization technique to minimize time-jerk trajectory planning under kinematic constraints through the application of a fifth-order B-spline curve. In [28], the authors interpolated the set points of the robot trajectory using a seventh-order B-spline interpolation curve, which was optimized through the SQP technique to produce time-optimal and jerk-continuous trajectories that satisfy nonlinear kinematic constraints. Furthermore, in [29] the authors used a seventh-degree B-spline curve interpolation, where smoothness is controlled via boundary conditions and motion constraints.

Some studies on robot trajectory planning have proposed Bézier curves [30–33]. In [30], the authors addressed the trajectory planning problem for a 6-DOF gluing robot. They used a Bézier curve to provide smooth and controlled tracking performance of the trajectory during the transition between two different movement segments in the Cartesian space. The optimal model for the least-time trajectory planning for the robot by a Bézier curve in the Cartesian space using the genetic algorithm (GA) is established in [32], which considers constraints like joint angular velocity and acceleration. Moreover, in [33], a method was used to get the trajectory of the rotational part of the robot manipulator. Since interpolating the transform matrix is difficult, the use of a quaternion is introduced, which is easily interpolated, and Bézier motion is applied to plan the smooth path of the end-effector.

The S-curve and AS-curve are discussed in several studies for the trajectory planning of robotic manipulators [34–39]. Lambrechts et al. [37] focused on an S-curve for single-axis motion control. It is demonstrated that these trajectories are time-optimal in the most relevant cases. Wu et al. [38] presented an algorithm to generate time-optimal and smooth joint trajectories in pick-and-place tasks, based on a locally enhanced asymmetrical jerk motion profile. Similarly, Perumaal and Jawahar [39] concentrated on the planning of a jerk-limited trigonometric S-curve trajectory for a 6-DOF robotic manipulator in pick-and-place tasks. Their simulation results reveal that this approach creates faster

and smoother movements compared with spline-based trajectories. Nevertheless, the maximum allowed jerk cannot be achieved in the acceleration and deceleration phases owing to the conservative nature of the synchronization technique, which results in unnecessary performance excess and an extra increase in the overall execution time.

Many papers in the literature investigate the combination of different functions for the description of trajectories [40–44]. Liu et al. [40] proposed to optimize the motion time by combining the cubic splines interpolation functions in the Cartesian space with the B-spline interpolation functions in the joint space. While cubic splines and B-splines functions simplify trajectory planning, they do not allow the robot manipulator to fully utilize its capabilities to save cycle time. Kucuk [41] took further steps to construct time-optimal trajectories by combining the cubic spline functions with the seventh-order polynomial and using a particle swarm optimization (PSO) algorithm for serial and parallel robotic manipulators. Other well-known approaches rely on polynomial interpolation functions, such as the third-order and fifth-order interpolation functions introduced by Cook and Ho [42]. This approach makes use of spline functions, although it does not ensure jerk continuity along the trajectory, whereas Petrínek et al. [43], as well as Boscaríol et al. [44], mix forth with fifth-order polynomial functions to interpolate the trajectory through a series of intermediate points and solve the problem of jerk continuity.

In addition to these various interpolation functions, in this paper special attention is paid to radial basis functions (RBFs) for interpolation, which are used effectively for scattered data interpolation problems in any number of dimensions. In this context, the work [45] presents two techniques for solving continuous-time optimal control problems based on RBF interpolation and arbitrary discretization. The choice of any global RBF as an interpolation function and of all arbitrary points as discretization points satisfy both the RBF collocation method and the RBF-Galerkin method, allowing for the development of a flexible solution to optimal control problems, particularly when involving non-smooth problems. This intriguing study might be thought of as an extension of [46]. Moreover, Mirinejad et al. [47] present a straightforward approach to optimum control problems based on the global radial basis function interpolation at any collocation point. The authors of [48] introduced a novel technique to get

smooth trajectory planning for robot manipulators in pick-and-place tasks by means of Gaussian RBF interpolation. Furthermore, Alipanah et al. [49] describe a numerical approach for solving the brachistochrone problem. It is based on the utilization of collocation points and the approximation of multiquadric radial basis function solutions. The approximate solution is computed in the form of a series whose components are easily calculated. To show the capabilities of the suggested technique, an error analysis and numerical results are presented. In [50], the MQ-RBF interpolation profile has been presented with application to the six-joint PUMA 560 manipulator to satisfy the null limit conditions on velocities and accelerations. However, that study overlooks the null limit for jerks at the start and end of the motion. Based on MQ-RBF to solve initial value problems, adaptive radial basis function methods have been developed in [51]. These adaptive methods use the free parameter in order to adaptively improve the local convergence of the numerical solution, and they are based on the regularity of the solution, which is given in derivatives of the solution, to control the rate of convergence.

This paper introduces a novel and efficient approach for constructing joint trajectories for manipulators based on multiquadrics-RBFs. By performing trajectory planning in the robot joint configuration, the approach proposed in this work enables easy achievement of transfers with via-points, while meeting imposed boundary conditions. Through trajectory optimization, the proposed technique yields minimum-time and smooth trajectory profiles that adhere to kinematic constraints of robotic manipulators. The main contributions of this work are summarized in the following:

- **Smooth trajectory planning based on MQ-RBFs:** The use of MQ-RBFs enables the planning of smooth trajectories for robot manipulators, providing continuous derivatives for any order within the interpolation interval and ensuring compliance to imposed kinematic constraints. Compared to other trajectory planning techniques, the proposed approach based on MQ-RBFs satisfies null limit conditions on velocities, accelerations, and jerks.
- **Optimal trajectory generation based on MQ-RBFs:** The proposed technique based on MQ-RBFs generates joint trajectories comparable to trigonometric spline and 5th-order B-spline methods in terms of transfer time, but with significantly

reduced jerk. Additionally, it overcomes issues associated with ill-conditioned matrices often encountered with high-degree curve spline interpolation methods.

- **Minimum jerk index:** Comparative analysis with state-of-the-art techniques demonstrates that the proposed MQ-RBF approach produces the minimum jerk index, while maintaining transfer time, showcasing its superiority over existing approaches.
- **Boundary conditions:** To the best of the authors' knowledge, very few interpolation functions in the literature address trajectory planning for robotic systems using the same interpolation technique such as MQ-RBFs, while allowing for variations in boundary conditions, such as setting null velocity, acceleration, and jerk, or null velocity and acceleration without null jerk.

The remainder of this paper is organized as follows. In Sect. 2, the issue of trajectory planning optimization is stated, namely the multi-objective function to optimize. In Sect. 3, MQ-RBF is chosen to create the joint trajectory, taking into consideration the boundary conditions. Section 4 outlines the applied optimization algorithms with the MQ-RBF process consideration. In Sect. 5, the suggested approach is contrasted with a variety of different RBF interpolation strategies. In Sect. 6, the proposed interpolation approach is tested to obtain a time-jerk optimal trajectory for the 6-DOF robot manipulator using two distinct optimization algorithms, SQP and NSGA-II. Furthermore, the experimental results on a UR5e robot are illustrated in Sect. 7. Finally, the conclusions of this work are reported in Sect. 8.

2 Formulation of the optimization problem

In industrial tasks, robotic systems need to move through a set of via-points in the operational space. Directly planning the trajectory in the Cartesian space can be challenging, so the above via-points are first converted into joint space coordinates through inverse kinematics. Based on these joint space via-points, a trajectory is designed and then optimized to achieve a specific goal. The trajectory is expected to have a travel time as low as feasible to maximize productivity in industrial applications, and it needs to be smooth to prevent excessive mechanical vibrations while imposing kinematic constraints. By keeping the travel time and

the motion smooth, the robot can operate efficiently and avoid issues like excessive vibration during its tasks.

Hence, the optimization problem is mathematically formulated as follows. The objective function that has to be minimized is:

$$F_{obj} = K_T T + K_J JM \tag{1}$$

The objective function in Eq. (1) is formulated using two conflicting objectives: travel time (T) and jerk (JM), each weighted by the factors K_T and K_J , respectively. The execution time is equal to:

$$T = \sum_{i=1}^{m-1} h_i \tag{2}$$

and the absolute mean jerk can be computed as:

$$JM = \sum_{j=1}^n \sqrt{\frac{1}{T} \int_0^T \ddot{q}_j^2(t) dt} \tag{3}$$

The optimization problem is subject to the following constraints:

$$\begin{cases} |\dot{q}_j(t)| \leq \dot{q}_j^{max}, & j = 1, \dots, n \\ |\ddot{q}_j(t)| \leq \ddot{q}_j^{max}, & j = 1, \dots, n \\ |\dddot{q}_j(t)| \leq \dddot{q}_j^{max}, & j = 1, \dots, n \end{cases} \tag{4}$$

The definitions of the aforementioned symbols can be found in Table 1. It is important to recognize that the objectives minimizing travel time and jerk (rate of change of acceleration) have conflicting effects. Minimizing travel time aims to make the robot complete its motion as quickly as possible, which often requires higher accelerations and jerks. Conversely, minimizing jerk ensures a smoother motion by reducing sudden changes in acceleration, but this generally increases the travel time because the robot must move more gradually. Thus, optimizing for one of these objectives often negatively affects the other, requiring a trade-off between travel time and smoothness in trajectory planning. Furthermore, the trajectory that effectively addresses the optimization problem must satisfy the interpolation conditions for all knots while also meeting the kinematic limits of velocity, acceleration, and jerk outlined in Eq. (4).

Table 1 Nomenclature of symbols

Symbol	Definition
F_{obj}	Objective function
K_T	Weight of the term proportional to the total time
T	Total time of the trajectory
K_J	Weight of the term proportional to the jerk
JM	Jerk index objective function
m	Number of interpolating points
n	Number of robot joints
h_i	Time interval between two knots
$\dot{q}_j, \ddot{q}_j, \ddot{\ddot{q}}_j$	Velocity, acceleration, and jerk of the j-th joint
$\dot{q}_j^{max}, \ddot{q}_j^{max}, \ddot{\ddot{q}}_j^{max}$	Velocity, acceleration, and jerk limits for the j-th joint

In order to evaluate the effectiveness of the MQ-RBF approach, the weighting coefficients K_T and K_J in the objective function expressed in Eq. (1) have to be properly selected. These coefficients balance the two contributions of the cost function: one term proportional to the total time, and one proportional to the jerk. The optimization problem can be solved for any values of K_T and K_J , which weight the cost function between the cases of minimum time and minimum jerk to achieve the desired trade-off between total time and smoothness of the trajectory. If $K_J = 0$, the objective function effectively reduces to minimizing only travel time (T), since the jerk term is no longer considered in the optimization. Conversely, if $K_T = 0$, the function focuses solely on minimizing jerk (JM), as the time-dependent term is removed. Thus, the selection of K_T and K_J determines the trade-off between minimizing time and achieving a smooth trajectory.

3 Proposed approach

3.1 Interpolation problem

The trajectory generation problem can be described as follows: given a set of points (x_i, y_i) for $i = 1, \dots, m$, which represent the via-points of a trajectory, the goal is to find a continuous function $f(x)$ that connects these

points as follows:

$$f(x_i) = y_i, \quad i = 1, \dots, m \quad \text{where} \quad x_i \neq x_j, \quad i \neq j \tag{5}$$

The trajectory is supposed to start at (x_1, y_1) and to end at (x_m, y_m) .

3.2 Definition of the trajectory by means of MQ-RBFs

To solve this problem, in this paper an original formulation based on the Multiquadric Radial Basis Functions (MQ-RBFs) technique is proposed for the function approximation. The MQ-RBF method was introduced in 1971 by R. L. Hardy [52,53] for fitting geographical data. The technique of MQ-RBF interpolation, which was developed by Hardy, received little attention until 1979. Nevertheless, subsequent research [54] showcased that the MQ-RBF technique outperforms other approaches in handling data interpolation issues across more than thirty distinct functions that were analyzed.

A radial basis function (RBF) is naturally defined as a multivariate function that can be represented as a function of a single variable. RBFs have been widely used in applications involving large data sizes, as they are well-suited for handling highly nonlinear problems. Moreover, an RBF is a real-valued function, meaning its value depends only on the radial distance from the origin or any specified reference point, known as the *center point* c , which is known as the radial basis function (RBF), so that:

$$\varphi(x, c) = \|x - c\| \tag{6}$$

Where φ is the RBF basis function, the norm $\| \cdot \|$ denotes the Euclidean distance between the coordinate of a point x and the RBF center c , with $r = \|x - c\|$. Any multivariate function that can be expressed as univariate function of the Euclidean norm $\| \cdot \|$ is called radial basis function. Some classical RBFs are shown in Table 2.

The RBF basis function can be:

- Infinitely smooth (global RBF) with a free positive parameter, often referred to as the shape parameter σ as in Gaussian and Multiquadric cases;

Table 2 Overview on classical RBFs

RBF Name	Symbol	Basis function
Gaussian	Ga-RBF	$\exp\left(-\frac{1}{2} \frac{r^2}{\sigma^2}\right), \sigma > 0$
Inverse multiquadric	IMQ-RBF	$1/\sqrt{r^2 + \sigma^2}, \sigma > 0$
Inverse quadrics	IQ-RBF	$1/(r^2 + \sigma^2), \sigma > 0$
Multiquadric	MQ-RBF	$\sqrt{r^2 + \sigma^2}, \sigma > 0$
Thin plate spline	TPS-RBF	$r^2 \log(r)$
Polyharmonic splines	PS-RBF	$\begin{cases} r^k & \text{if } k = 1, 3, 5, \dots \\ r^k \ln(r) & \text{if } k = 2, 4, 6, \dots \end{cases}$

- Piecewise smooth without a shape parameter σ , such as in thin plate splines and polyharmonic splines.

In this study, particular focus is given to robot trajectory interpolation using MQ-RBF, which is expressed as follows [55]:

$$\varphi(r) = \sqrt{r^2 + \sigma^2} \quad \text{with } \sigma > 0 \tag{7}$$

A significant property of MQ-RBF is that it is nonsingular, simple to use [56], and a conditionally positive definite function [57]. Furthermore, the user should set the shape parameter σ , which is utilized to modify the overall form of the RBF during the approximation process [54].

3.2.1 Velocity profile expression

The velocity profile expression was determined using an analytical derivation of Eq. (7), yielding the following formula:

$$\frac{\partial \varphi(r)}{\partial x} = \frac{(x - c)}{\sqrt{r^2 + \sigma^2}} = \frac{(x - c)}{\varphi(r)} \tag{8}$$

3.2.2 Acceleration profile expression

The acceleration profile expression was determined using an analytical derivation of Eq. (8), yielding the following formula:

$$\begin{aligned} \frac{\partial^2 \varphi(r)}{\partial x^2} &= \frac{1}{\sqrt{r^2 + \sigma^2}} - \frac{(x - c)^2}{(r^2 + \sigma^2)^{3/2}} \\ &= \left(1 - \frac{(x - c)^2}{\varphi(r)^2}\right) \frac{1}{\varphi(r)} \end{aligned} \tag{9}$$

3.2.3 Jerk profile expression

The jerk profile expression was determined using an analytical derivation of Eq. (9), yielding the following formula:

$$\begin{aligned} \frac{\partial^3 \varphi(r)}{\partial x^3} &= \frac{3(x - c)^3}{(r^2 + \sigma^2)^{5/2}} - \frac{3(x - c)}{(r^2 + \sigma^2)^{3/2}} \\ &= \left(\frac{3(x - c)^3}{\varphi(r)^2} - 3(x - c)\right) \frac{1}{\varphi(r)^3} \end{aligned} \tag{10}$$

Thus, to approximate the interpolating function f everywhere by a linear combination of certain RBFs φ_j , yields, [55]:

$$f(x) = \sum_{j=1}^m \omega_j \varphi_j(r_j) \quad \text{with } r_j = \|x - c_j\| \tag{11}$$

Where $f(x)$ is a sum of m RBFs φ_j , each associated with a center $c_j = x_j$ and weighted by an appropriate weight ω_j . x represents the input for which the function f has to be evaluated. By choosing m interpolate nodes, the function $f(x)$ can be approximated as:

$$y_i = \sum_{j=1}^m \omega_j \varphi_j \|x_i - x_j\| \quad i = 1, \dots, m \tag{12}$$

After some algebra, solving the interpolation problem leads to a system of linear equations in the form:

$$AW = Y \tag{13}$$

where Y is a vector with elements y_i , W is a vector with elements ω_j , and A is a matrix with rows $\varphi(x_i)$. Note that the entries of the interpolation matrix are given by

$A_{ij} = \varphi_j(x_i)$, with $i, j = 1, \dots, m$. Thus, using Eq. (7), the coefficient matrix can be defined as follows:

$$A = \begin{bmatrix} 1 & \varphi_2(x_1) & \cdots & \varphi_m(x_1) \\ \varphi_1(x_2) & 1 & \cdots & \varphi_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_m) & \varphi_2(x_m) & 1 & \cdots & 1 \end{bmatrix} \tag{14}$$

The associated weights W may be determined by solving Eq. (13) as follows:

$$W = A^{-1}Y \tag{15}$$

To achieve this objective, three virtual via-points are added near x_1 , and three virtual via-points near x_m , so that the initial and final conditions for velocity, acceleration, and jerk can be respected. Thus, Eq. (12) becomes:

$$f(x) = \sum_{j=1}^{m+6} \omega_j \varphi_j \|x - c_j\| \tag{19}$$

The associated linear system is defined by the following elements: the coefficients matrix A , the vector of unknown weight coefficients W , and the vector Y . The coefficients matrix A with dimensions $[m + 6 \times m + 6]$ is defined as:

$$A = \begin{bmatrix} 1 & \varphi_2(x_1) & \cdots & \varphi_m(x_1) & \varphi_{m+1}(x_1) & \cdots & \varphi_{m+6}(x_1) \\ \varphi_1(x_2) & 1 & \cdots & \varphi_m(x_2) & \varphi_{m+1}(x_2) & \cdots & \varphi_{m+6}(x_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \varphi_1(x_m) & \varphi_2(x_m) & \cdots & 1 & \varphi_{m+1}(x_m) & \cdots & \varphi_{m+6}(x_m) \\ \varphi'_1(x_{m+1}) & \varphi'_2(x_{m+1}) & \cdots & \varphi'_m(x_{m+1}) & \varphi'_{m+1}(x_{m+1}) & \cdots & \varphi'_{m+6}(x_{m+1}) \\ \varphi'_1(x_{m+2}) & \varphi'_2(x_{m+2}) & \cdots & \varphi'_m(x_{m+2}) & \varphi'_{m+1}(x_{m+2}) & \cdots & \varphi'_{m+6}(x_{m+2}) \\ \varphi''_1(x_{m+3}) & \varphi''_2(x_{m+3}) & \cdots & \varphi''_m(x_{m+3}) & \varphi''_{m+1}(x_{m+3}) & \cdots & \varphi''_{m+6}(x_{m+3}) \\ \varphi''_1(x_{m+4}) & \varphi''_2(x_{m+4}) & \cdots & \varphi''_m(x_{m+4}) & \varphi''_{m+1}(x_{m+4}) & \cdots & \varphi''_{m+6}(x_{m+4}) \\ \varphi'''_1(x_{m+5}) & \varphi'''_2(x_{m+5}) & \cdots & \varphi'''_m(x_{m+5}) & \varphi'''_{m+1}(x_{m+5}) & \cdots & \varphi'''_{m+6}(x_{m+5}) \\ \varphi'''_1(x_{m+6}) & \varphi'''_2(x_{m+6}) & \cdots & \varphi'''_m(x_{m+6}) & \varphi'''_{m+1}(x_{m+6}) & \cdots & \varphi'''_{m+6}(x_{m+6}) \end{bmatrix} \tag{20}$$

3.2.4 Boundary conditions

The time scale in the context of robot trajectory is denoted by x as an independent variable, and the points need to be interpolated in an ascending order, i.e., $x_1 < x_2 < \dots < x_m$. Similarly, the velocity, acceleration, and jerk associated with the first and last knots of a trajectory have to be assigned, i.e.,

$$v_1 = \frac{\partial f(x = x_1)}{\partial x}, \quad v_m = \frac{\partial f(x = x_m)}{\partial x} \tag{16}$$

$$a_1 = \frac{\partial^2 f(x = x_1)}{\partial x^2}, \quad a_m = \frac{\partial^2 f(x = x_m)}{\partial x^2} \tag{17}$$

$$j_1 = \frac{\partial^3 f(x = x_1)}{\partial x^3}, \quad j_m = \frac{\partial^3 f(x = x_m)}{\partial x^3} \tag{18}$$

where velocities v_1, v_m , accelerations a_1, a_m , and jerks j_1, j_m are given data of the problem.

Regarding the matrix A , the last six rows are calculated using the Eqs. (8) and (9). Moreover, the extra knots of RBF trajectory are introduced in the following instances:

$$\begin{aligned} x_{m+1} &= x_1 + \delta(x_2 - x_1), & x_{m+2} &= x_m - \delta(x_m - x_{m-1}) \\ x_{m+3} &= x_1 + 2\delta(x_2 - x_1), & x_{m+4} &= x_m - 2\delta(x_m - x_{m-1}) \\ x_{m+5} &= x_1 + 3\delta(x_2 - x_1), & x_{m+6} &= x_m - 3\delta(x_m - x_{m-1}) \end{aligned} \tag{21}$$

where $\delta = 0.02$ represents a control parameter ensuring smooth transitions between virtual via-points.

The vector of unknown weight coefficients W with dimension $[m + 6 \times 1]$ is:

$$W = [\omega_1, \dots, \omega_m, \omega_{m+1}, \dots, \omega_{m+6}]^T \tag{22}$$

The last element of the linear system Y with dimension $[m + 6 \times 1]$ becomes:

$$Y = [y_1, \dots, y_m, v_1, v_m, a_1, a_m, j_1, j_m]^T \quad (23)$$

4 Proposed optimization techniques

This section provides the definitions of the optimization techniques, as well as the optimization process considered for trajectory generation using MQ-RBF profiles.

4.1 Optimization process consideration

The optimization problem at hand focuses on determining the minimum-time smooth motion trajectories for robotic manipulators. In this approach, the shape parameters of each trajectory candidate based on MQ-RBF are determined by unique parameters chosen for each joint trajectory, denoted as σ_i . These parameters are adjusted during the optimization process. The optimization variable vector, denoted as $F = (h_1, \dots, h_{m-1}, \sigma_1, \dots, \sigma_n)$, represents the solution to the problem. The objective is to find an optimal set of values for the time intervals h_j with $j = 1, \dots, m - 1$, and suitable shape parameters σ_i with $i = 1, \dots, n$ that minimize the cost function while adhering to the kinematic constraints for all joints. This optimization process utilizes the Multiquadric-RBF interpolation technique, ensuring smooth trajectories while satisfying the imposed kinematic constraints.

4.2 Optimization framework

The optimization framework is designed to address the multi-objective nature of the problem, focusing on both trajectory time and jerk minimization. Each optimization technique is applied independently, ensuring a thorough exploration and exploitation of the solution space.

The Sequential Quadratic Programming (SQP) is a powerful optimization method suitable for solving non-linear optimization problems with constraints. While it is not inherently multi-objective, it can be used for precise single-objective optimization after converting the multi-objective problem into a single-objective one. The SQP algorithm iteratively solves a sequence of

quadratic programming subproblems to converge to an optimal solution. It is ideal for detailed optimization, particularly for problems with complex constraints, and it solves iteratively quadratic approximations of the original problem. The SQP method effectively manages kinematic constraints such as velocity, acceleration, and jerk, enhancing solution precision through iterative refinement.

The Elitist Non-dominated Sorting Genetic Algorithm II (NSGA-II) is a popular evolutionary algorithm used for solving multi-objective optimization problems. It effectively maintains a balance between exploration and exploitation by preserving a diverse set of solutions and promoting convergence towards the Pareto front. NSGA-II handles conflicting objectives efficiently by finding a set of optimal trade-off solutions [58,59].

The proposed optimization algorithm employs SQP and NSGA-II as independent techniques to address the trajectory planning problem. NSGA-II explores the solution space by maintaining diversity and promoting convergence towards the Pareto front, while SQP refines solutions to enhance precision within the constraints. By applying NSGA-II and SQP separately, the algorithm ensures a comprehensive optimization process, resulting in smooth and efficient trajectories that meet all kinematic constraints. This dual-application strategy allows for thorough exploration and exploitation of the solution space, ultimately leading to the generation of high-quality trajectories for robotic manipulators.

5 Trajectory planning using RBFs interpolation

To provide an illustrative example, the trajectory planning for specified tasks is examined using the same single link trajectory, as shown in [20]. The trajectory must establish connections between the joint positions $q = [120^\circ, 60^\circ, 80^\circ, 120^\circ, 0^\circ]$ while simultaneously adhering to further given constraints at both the starting and ending positions of motion, including the null limit velocity, acceleration, and jerk. The time intervals used are equal to 2s between two consecutive knots. This example aims at examining the application of MQ-RBF interpolation approach for planning joint trajectories for manipulators and compare the yielded results with those achieved by the Ga-RBF, IMQ-RBF, and IQ-RBF trajectory models. To achieve this objective, three

Fig. 1 Trajectories obtained using four different RBF interpolation approaches: joint positions (a), velocities (b), accelerations (c), and jerks (d)

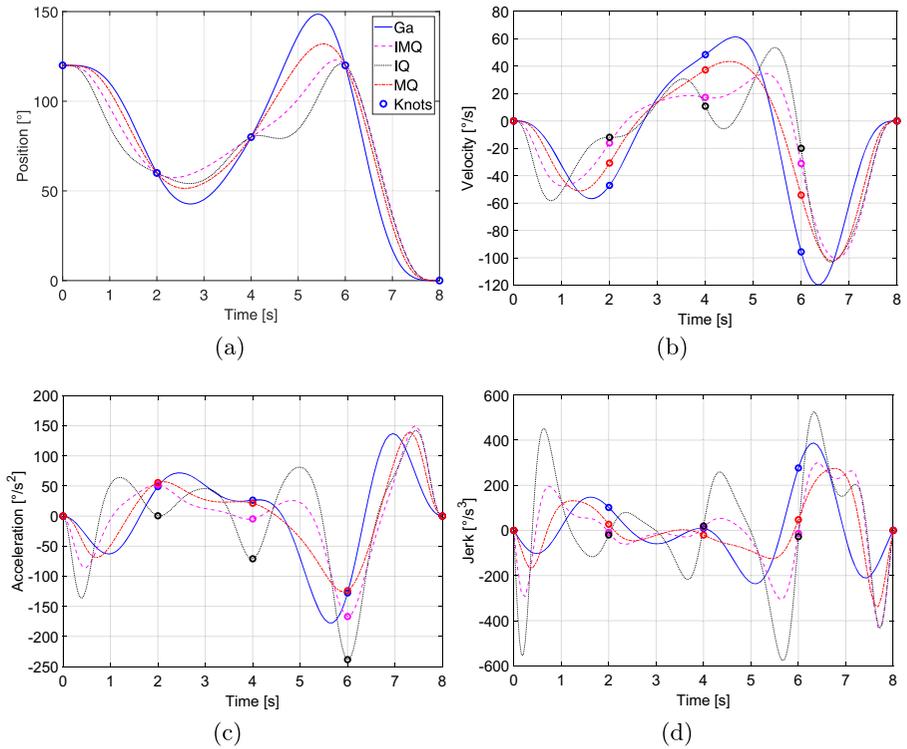


Table 3 Maximum values of position Q_{max} , velocity V_{max} , acceleration A_{max} , and jerk J_{max} obtained with different RBFs

RBF	Q_{max} [deg]	V_{max} [deg/s]	A_{max} [deg/s ²]	J_{max} [deg/s ³]
Ga-RBF	148.53	119.58	177.55	387.59
IMQ-RBF	122.88	100.45	166.97	429.95
IQ-RBF	120.85	103.01	238.54	577.20
MQ-RBF	131.99	102.43	138.88	338.47

extra knots are introduced to the limit configurations according to Eq. (21). Furthermore, the shape parameter σ is considered to be uniform across all applied RBF models, and it is supposed to be $\sigma = 1$.

The trajectories plots of the four RBF models that connect the via-points and their related derivatives profiles are given in Fig. 1 for comparison. Table 3 summarizes the maximum recorded values. From Fig. 1, it can be noted that the four RBF position profiles appear not to be considerably different, and they connect all via-points positions in a smooth way. From the derivatives profiles of the trajectory, it is apparent that the Ga-RBF and the IMQ-RBF trajectories behave like the MQ-RBF trajectory and that the MQ-RBF model has a better score in terms of maximum values of jerk when compared to other RBF models, as stated in Table 3.

Furthermore, Fig. 1 shows that the IQ-RBF trajectory has a superior displacement profile. Nevertheless, this advantage is achieved by higher amounts of velocity and acceleration and, consequently, an oscillating jerk trajectory profile with much greater values in contrast to the other RBF models.

In addition, the computation of the trajectory parameters takes an average of 0.48 s for the four RBF interpolation methods, and 6.61 ms for the MQ-RBF trajectory method on a personal computer with an Intel Core i7-5850U 2.00 GHz processor and 8 GB RAM in Matlab™ environment.



Fig. 2 The Puma 560 robot manipulator

6 Trajectory generation for 6-DOF robot manipulator

In this second example, the goal is to generate a smooth motion for a 6-DOF robot manipulator PUMA 560 (Fig. 2) using the proposed MQ-RBF interpolation approach, and to evaluate its performance by comparing the outcomes with those obtained using other interpolation functions proposed in recent studies. The evaluation is conducted through two optimization techniques: first, a comparison with fifth-order B-splines and trigonometric spline interpolation functions, as presented in [23,60], respectively, where the SQP optimization technique was applied; and second, a comparison with fifth-order B-splines functions, as shown in [27], where the trajectories were optimized using the NSGA-II optimization technique.

There are four different poses ($m = 4$) of the manipulator at each knot point. In this paper, joint space values obtained from one of the acceptable inverse kinematics solutions have been directly used, as described by the above authors. The joint values yielded for these knots are presented in Table 4, while meeting the kinematic limits reported in Table 5.

It should be noted that the initial and final values of the joint trajectory were set to zero for the first, second, and third derivatives of trajectory profiles as in [23,27,60]. Thus, to enable the comparison of the results, three additional knots with the associated Multiquadric-RBFs have been added in each extreme interval (h_1 and h_n), as described in Sect. 3.2.4. Therefore, a linear system of $(m+6)$ equations must be solved to obtain the Multiquadric-RBF weight coefficients.

6.1 Smooth trajectory using SQP technique

In this case, to ensure a fair comparison of outcomes, the values of the weighting coefficients K_T and K_J in the objective function Eq. (1) have been adjusted, as prescribed by Gasparetto and Zanotto [23] and Simon-Isik [60]. This adjustment has been made so that the travel time of the proposed MQ-RBF approach and the technique presented in [23,60] are identical, which is approximately 9.1 s. By implementing the MQ-RBF approach, the total travelling time of the trajectory was found to be $T = 9.0971$ s. Furthermore, the optimal values of time intervals are $h_1 = 3.4197$ s, $h_2 = 2.3798$ s, and $h_3 = 3.2976$ s. In addition, the optimal values of MQ-RBF shape parameters are $\sigma_1 = 1.5312$, $\sigma_2 = 1.6210$, $\sigma_3 = 1.5216$, $\sigma_4 = 1.8624$, $\sigma_5 = 1.6740$, and $\sigma_6 = 1.5347$.

Figure 3 shows the joint trajectories and their corresponding third-order derivatives of the 6-DOF robot manipulator. The graphs in the figure indicate that the initial and final points of the joint trajectory curves have zero values for the first, second, and third derivatives, which confirms that all kinematic limits have been met.

Table 6 presents the maximum kinematic values for all joints. The values in the table show that all kinematic requirements for the robot trajectory have been fulfilled. Furthermore, these results are compared with those obtained in Gasparetto's and Simon's works in [23,60]. To minimize the time-jerk objective function, Gasparetto [23] used fifth-order B-splines interpolation and the SQP optimization technique. Simon's work [60], on the other hand, uses a minimum-jerk trigonometric spline to interpolate the trajectory. As indicated in Table 6, the employment of the MQ-RBF approach resulted in lower jerk values for joints 1, 3, 4, 5, and 6 compared to Gasparetto and Simon methods, with reduction rates ranging from 6% to 11% and 5% to 21% compared with those obtained by the algorithms proposed in [23,60], respectively.

The mean kinematic values of velocities, accelerations, and jerks for all joints obtained using the MQ-RBF method are presented in Table 7 and compared again with the values obtained using the techniques proposed by Gasparetto [23] and Simon [60]. The comparison reveals that the proposed MQ-RBF interpolation method produces comparable results to the outcomes provided by the techniques of Gasparetto and Simon's techniques in [23,60] concerning the lower mean val-

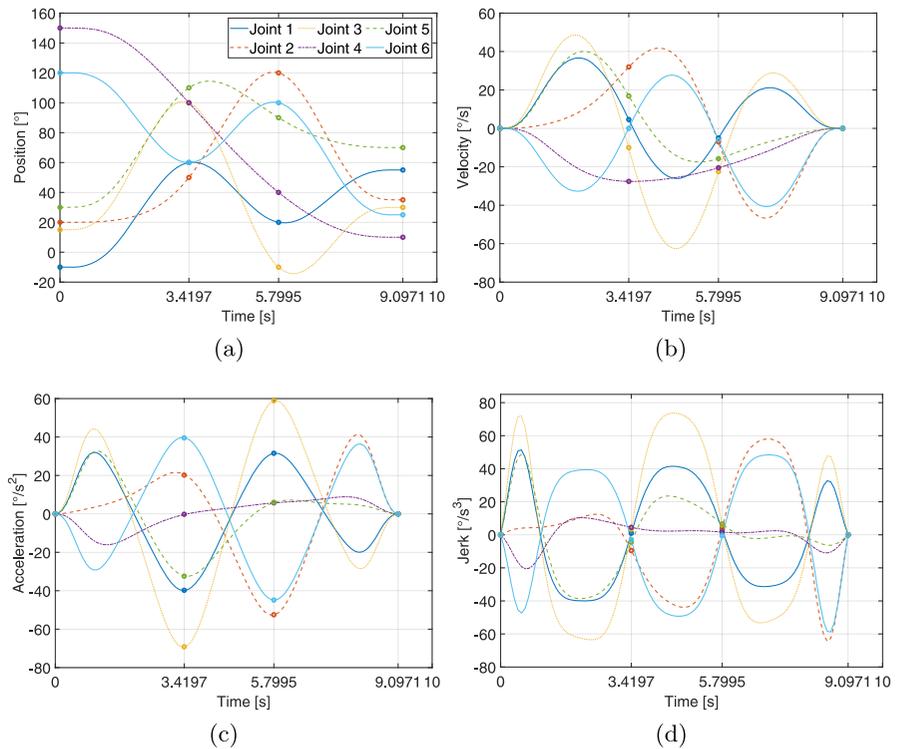
Table 4 Joint-space knots for the trajectory planning

Knot	Joint [deg]					
	q_1	q_2	q_3	q_4	q_5	q_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

Table 5 Kinematic limits of the PUMA robot joints

Limit	Joint [deg]					
	q_1	q_2	q_3	q_4	q_5	q_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. 3 Optimal joint positions (a), velocities (b), accelerations (c), and jerks (d) using SQP



ues for velocity, acceleration, and jerk of almost all the robot joints.

6.2 Smooth trajectory using NSGA-II technique

To further test the proposed approach, in this section a second case is presented, where the MQ-RBF interpolation method is compared with the fifth-order B-splines interpolation function, in which the stated problem with

Table 6 Maximum kinematic values obtained using the SQP optimization technique

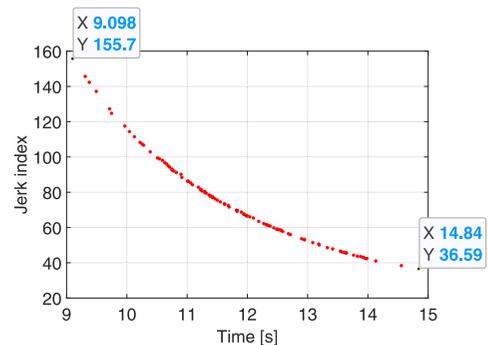
Algorithm		q_1	q_2	q_3	q_4	q_5	q_6
Proposed approach	V_{max} [$^\circ/s$]	36.66	46.67	62.59	27.60	40.02	40.71
	A_{max} [$^\circ/s^2$]	39.76	52.67	69.38	16.09	32.70	44.84
	J_{max} [$^\circ/s^3$]	51.35	64.50	73.67	20.49	48.84	58.72
Gasparetto [23]	V_{max} [$^\circ/s$]	37.57	41.96	61.49	28.90	41.53	38.14
	A_{max} [$^\circ/s^2$]	39.07	43.65	65.60	15.87	33.88	39.94
	J_{max} [$^\circ/s^3$]	54.94	65.90	78.13	23.02	52.73	65.15
Simon [60]	V_{max} [$^\circ/s$]	39.84	47.67	57.54	28.67	44.29	43.00
	A_{max} [$^\circ/s^2$]	39.03	46.40	62.05	19.89	35.61	43.73
	J_{max} [$^\circ/s^3$]	54.82	59.28	80.84	25.99	49.04	61.47

Table 7 Mean kinematic values obtained using the SQP optimization technique

Algorithm		q_1	q_2	q_3	q_4	q_5	q_6
Proposed approach	V_{mean} [$^\circ/s$]	15.92	20.24	26.68	15.23	14.03	19.13
	A_{mean} [$^\circ/s^2$]	18.26	19.26	30.50	6.01	12.53	22.02
	J_{mean} [$^\circ/s^3$]	26.86	25.17	43.79	5.43	15.75	32.71
Gasparetto [23]	V_{mean} [$^\circ/s$]	16.94	20.70	27.45	15.38	15.92	19.47
	A_{mean} [$^\circ/s^2$]	19.08	18.15	30.60	6.50	14.92	21.84
	J_{mean} [$^\circ/s^3$]	26.26	20.69	41.26	6.39	18.18	30.35
Simon [60]	V_{mean} [$^\circ/s$]	16.87	22.11	26.41	15.38	16.76	20.24
	A_{mean} [$^\circ/s^2$]	19.48	19.99	30.07	6.53	15.02	23.52
	J_{mean} [$^\circ/s^3$]	27.51	23.97	41.27	9.10	18.07	33.77

optimum time-jerk trajectories (Eq. 2 and 3) is a multi-objective nonlinear problem solved by using a powerful optimization algorithm. The NSGA-II technique in MatlabTM was applied to optimize the objective functions for the robot manipulator using the same control parameters as in [27], where the population size is 100, the generation number is 80, and the mutation probability is selected as $1/n = 1/5 = 0.2$ (where n represents the number of variables) as proposed by Deb [61]. Distribution indexes for real-coded mutation and crossover operators are 100 and 20, respectively, as recommended by Deb [61].

Figure 4 shows the optimum Pareto front derived using NSGA-II. On this front, the transfer time varies from 9.098 to 14.84 s, and the jerk index ranges from 36.58 to 155.70 $^\circ/s^3$. In contrast, the Huang's approach [27] employs 5th-order B-splines to interpolate the trajectory and optimize it using NSGA-II. In this study, the transfer time varies from 9.058 to 13.96 s, and the jerk index ranges from 55.55 to 188.98 $^\circ/s^3$. The afore-

**Fig. 4** Pareto front for time-jerk optimal trajectory planning for the 6-DOF robot manipulator

mentioned Pareto front was also capable of producing a solution with the same transfer time of about 9.1 s. The time intervals minimized by MQ-RBF interpolation approach solution in this work ($h_1 = 3.4315$ s, $h_2 = 2.3228$ s, $h_3 = 3.3438$ s, $T = 9.0981$ s) require

only a jerk index of $155.70^\circ/s^3$. While in Huang's work [27], for the same transfer time, a jerk index of $188.98^\circ/s^3$ is needed, which is a considerable reduction of about 17%. Furthermore, Fig. 4 shows that the solution with a transfer time of about 14.84 s has a jerk index of $36.58^\circ/s^3$, representing a 34% improvement above [27].

The trajectories of the robot manipulator and their derivatives corresponding to the minimum time-jerk solution are illustrated in Fig. 5. The results in this figure indicate that the first three derivatives of joint trajectory curves are null at the start and end points of the trajectory and also show that they meet all kinematic limits of the manipulator. Furthermore, Table 8 illustrates the maximum kinematic values for all joints. These values demonstrate that all kinematic limits of the manipulator have been satisfied, and the maximum jerk values of joints 1, 3, 4, 5, and 6 produced by the MQ-RBF approach are lower by 3%, 4%, 34%, 21%, and 3%, respectively, compared to those yielded by the fifth-order B-Spline functions [27]. Additionally, as indicated in Table 9, the mean jerk values of joints 1, 5, and 6 are lower by 3%, 9%, and 1%, respectively, than those of [27].

6.3 Discussion and comparison of results

To conduct a complete comparison of MQ-RBF considering both optimization techniques, Table 10 reports the average kinematic values for the 6-joint robot manipulator. These values illustrate the outcomes of employing the MQ-RBF approach with both optimization techniques, which are as follows:

- For the SQP optimization technique, the maximum jerk value yielded by the MQ-RBF approach resulted in lower jerk values, with reduction rates ranging of 6% and 4% compared to those generated by the algorithms described in [23, 60], respectively. Moreover, while the mean jerk value achieved by [23] is 4% lower than the proposed approach, the MQ-RBF method still yields lower jerk values compared to the trajectory interpolated by trigonometric splines in Simon [60]. From Fig. 6, it can be concluded that the MQ-RBF interpolation method yields competitive results compared to the previous studies [23, 60] in terms of kinematic values.

- For the NSGA-II optimization technique, the maximum and mean jerk values obtained by the MQ-RBF approach are lower compared to those yielded by fifth-order B-Splines functions [27]. Particularly notable is the reduction in maximum jerk values, with a reduction rate ranging from 7%. From Fig. 6, it can be seen that the proposed approach outperformed the previous studies [27] in terms of velocity, acceleration, and jerk values.

7 Experimental results

To verify the practical usability of the proposed interpolation approach on a real robotics scenario, the method is experimentally tested on a 6-DOF UR5e robot manufactured by Universal Robots using the example trajectory shown in the previous section. The robot, shown in Fig. 7, is a lightweight and small-sized industrial arm with a payload of 5 kg, a weight of 20.6 kg, and a maximum reach of 850 mm. The manipulator is programmed in ROS (Robot Operating System) Melodic Morenia using Python and it is controlled with a computer equipped with 32 GB RAM and an Intel Core i9 processor running Ubuntu 18.04. Data acquisition from the robot is performed by means of the Real-Time Data Exchange (RTDE) protocol through a TCP/IP connection.

To ensure compliance with the robot operational constraints and prevent collisions with the workbench, adjustments were made to the test task described in the previous section. The experimental validation was performed considering all the six joints of the UR5e robot, with modifications involved inverting the sign of the knot positions for joint 2, 3 and 4, while maintaining their displacements. The modified knots positions of joint 2, 3 and 4 were set as follows $[-20^\circ, -50^\circ, -120^\circ, -35^\circ]$, $[-15^\circ, -100^\circ, 10^\circ, -30^\circ]$, and $[-150^\circ, -100^\circ, -40^\circ, -10^\circ]$ with an optimal transfer time of 9.0971 s. The robotic arm operates with a positioning accuracy of 0.1 mm. The considered kinematic limits are reported in Table 5.

The measured joint-space profiles for all six robot arms are presented in Fig. 8, providing a comparison between experimental and simulation results. The results indicate that the robot successfully tracked the trajectory with accuracy, maintaining smooth joint motion throughout. Overall, the experimental profiles are in accordance with the simulated ones in general.

Fig. 5 Optimal joint positions (a), velocities (b), accelerations (c), and jerks (d) using NSGA-II

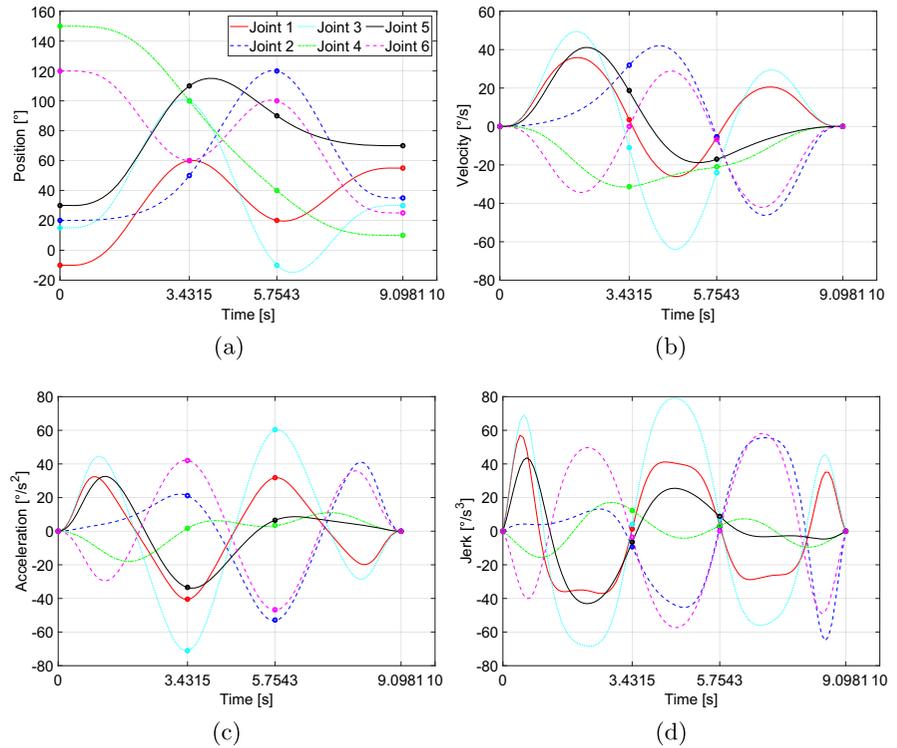


Table 8 Maximum kinematic values obtained using the NSGA-II optimization technique

Algorithm		q_1	q_2	q_3	q_4	q_5	q_6
Proposed approach	V_{max} [$^{\circ}/s$]	35.85	46.25	63.96	31.45	41.12	42.07
	A_{max} [$^{\circ}/s^2$]	40.45	53.29	71.18	17.94	33.88	46.94
	J_{max} [$^{\circ}/s^3$]	57.06	65.16	79.14	17.08	43.63	58.11
Huang [27]	V_{max} [$^{\circ}/s$]	39.27	47.05	50.65	29.15	43.35	41.98
	A_{max} [$^{\circ}/s^2$]	42.67	53.04	70.53	17.6	35.75	48.19
	J_{max} [$^{\circ}/s^3$]	59.00	62.28	82.92	26.20	55.4	60.20

Table 9 Mean kinematic values obtained using the NSGA-II optimization technique

Algorithm		q_1	q_2	q_3	q_4	q_5	q_6
Proposed approach	V_{mean} [$^{\circ}/s$]	15.92	20.18	26.80	15.23	14.15	19.15
	A_{mean} [$^{\circ}/s^2$]	17.97	19.22	31.11	6.84	13.05	22.94
	J_{mean} [$^{\circ}/s^3$]	27.15	25.35	44.57	7.05	16.34	33.67
Huang [27]	V_{mean} [$^{\circ}/s$]	16.33	21.50	26.33	15.39	15.59	19.78
	A_{mean} [$^{\circ}/s^2$]	19.36	20.74	30.93	6.41	14.61	23.51
	J_{mean} [$^{\circ}/s^3$]	27.86	25.33	43.61	7.00	17.93	33.95

Table 10 Average of maximum and mean kinematic values using SQP and NSGA-II

Algorithm	Optimiz. approach	Interp. approach		Av. of max. kin. values	Av. of mean kin. values	
Proposed approach	SQP	MQ-RBF	$V [^\circ/s]$	42.37	18.54	
			$A [^\circ/s^2]$	42.58	18.10	
			$J [^\circ/s^3]$	52.93	24.95	
Gasparetto [23]	SQP	Quintic	$V [^\circ/s]$	41.59	19.31	
			B-splines	$A [^\circ/s^2]$	39.66	18.51
				$J [^\circ/s^3]$	56.64	23.85
Simon [60]	SQP	Trigonometric splines	$V [^\circ/s]$	43.50	19.62	
			$A [^\circ/s^2]$	41.11	19.10	
			$J [^\circ/s^3]$	55.24	25.61	
Proposed approach	NSGA-II	MQ-RBF	$V [^\circ/s]$	43.45	18.57	
			$A [^\circ/s^2]$	43.95	18.52	
			$J [^\circ/s^3]$	53.36	25.69	
Huang [27]	NSGA-II	Quintic	$V [^\circ/s]$	41.90	19.15	
			B-splines	$A [^\circ/s^2]$	44.63	19.26
				$J [^\circ/s^3]$	57.66	25.94

Fig. 6 Average of maximum and mean kinematic values using SQP (a, b) and NSGA-II (c, d)

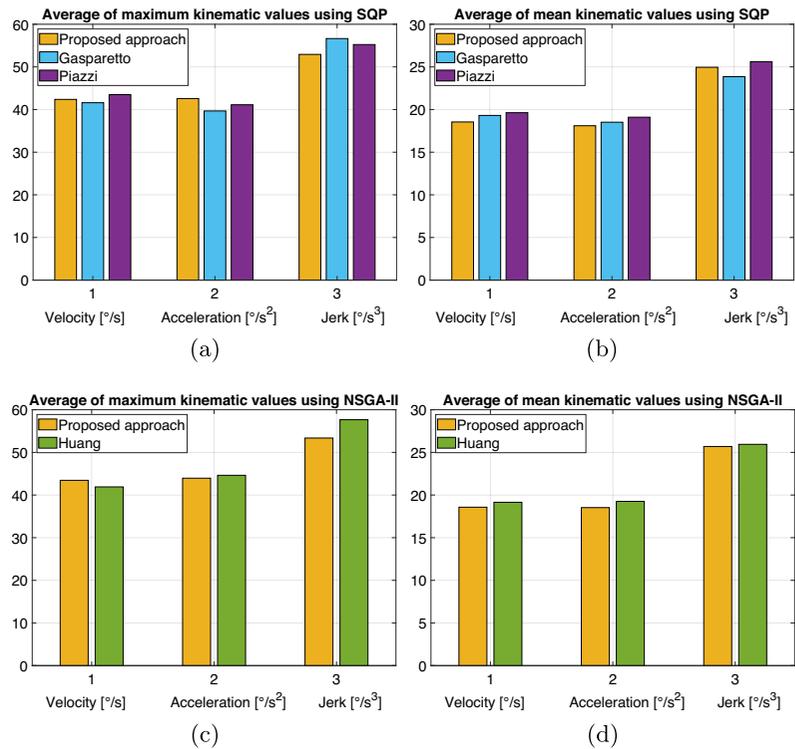


Fig. 7 The UR5e manipulator by Universal Robots and its control box (a); path of the robot end-effector in the Cartesian space (b)

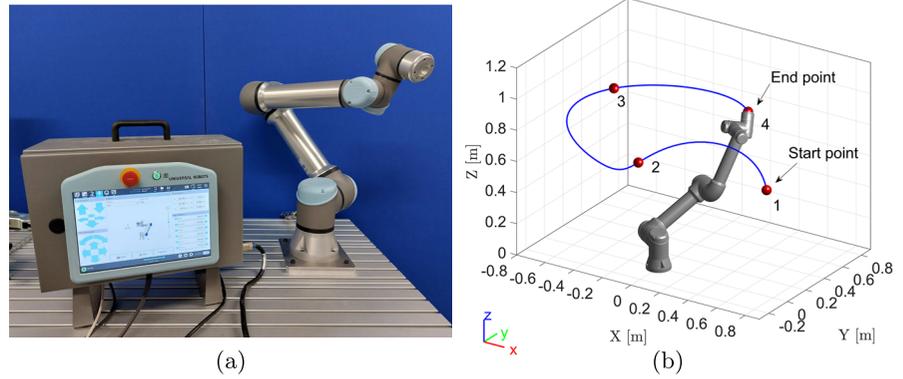
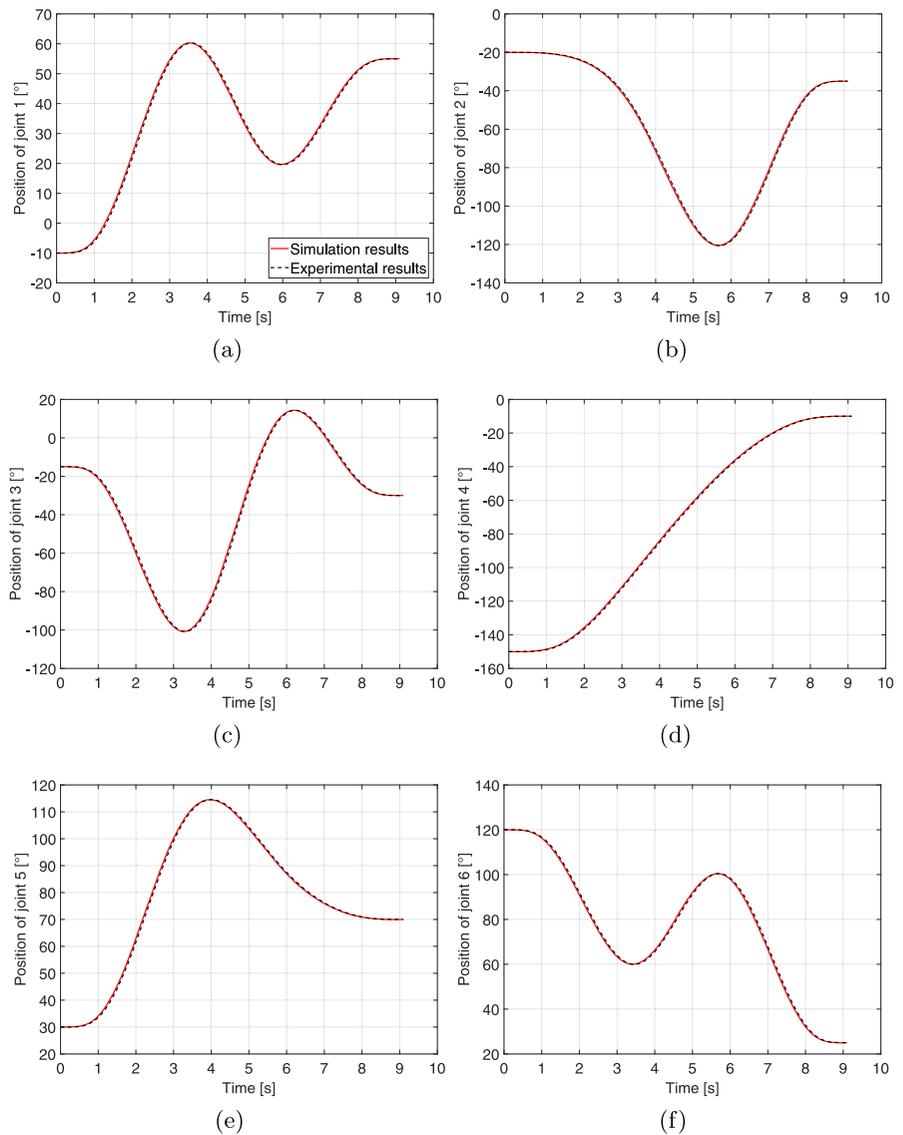


Fig. 8 Desired and experimental joint positions of the UR5e robot using the proposed approach



Although tiny deviation exists between the experimental curves and the simulated ones, the range and the variation tendency of the curves in each figure are nearly the same. The maximum position errors for joints one through six are around 1.3° , 1.7° , 2.3° , 1.0° , 1.5° , and 1.5° , respectively. At the end of the trajectory, the errors for all joints arm remain within 0.01° . Based on these observations the experimental results confirm the practical effectiveness of the proposed trajectory planning algorithm. The optimized trajectory was indeed successfully executed on a real robot, validating the applicability and reliability of the proposed MQ-RBF interpolation approach.

8 Conclusions

A novel and efficient approach for designing joint trajectories for robotic manipulators based on Multiquadric-RBFs has been presented in this work. The trajectory planning is performed in the robot joint space, where it is possible to easily obtain motions between pre-defined via-points that meet the imposed boundary conditions. By optimizing the trajectory, a minimum-time and smooth trajectory profile that meets the kinematic constraints is obtained. The proposed technique has been validated through both simulation and experimental tests, and its results have been compared with those of previous state-of-the-art approaches [20,23,27,60].

The performance of the proposed technique can be summarized based on the following observations: the construction of the trajectory by using MQ-RBFs enables to plan smooth trajectories for the robotic manipulator while meeting imposed kinematic constraints and satisfying not only the null limit conditions on velocities and accelerations, but also the null limit conditions on velocities, accelerations, and jerks compared to other trajectory planning techniques. The proposed technique based on MQ-RBFs has shown the ability to plan joint trajectories comparable to trigonometric splines and 5th-order B-splines in terms of transfer time, while offering an infinite order of derivation and reduced jerk. Furthermore, the proposed approach has advantages over spline interpolation, particularly in reducing the risk of ill-conditioned matrices with high-degree curves. Finally, compared to previous techniques, the MQ-RBFs approach achieves a lower jerk index, while maintaining the transfer time. However, one limitation of the proposed approach is that the

average of the maximum kinematic values, specifically velocity and acceleration, is not always lower than that of compared approaches in the considered test cases.

In future developments of this work, the proposed MQ-RBF approach will be applied for the planning of optimal smooth trajectories for mobile robots and parallel manipulators. Furthermore, future developments of this work will also consider the extension of the proposed approach to higher-order derivatives, e.g., jounce, to further improve the performance of the approach in planning optimal smooth trajectories. Future works will also consider different optimization strategies and alternative RBF-hybrid methods to generate smooth trajectories for robotic manipulators.

Acknowledgements The authors would like to thank Mr. Giuliano Fabris for the help with the experimental setup.

Author contributions NB conducted the algorithm design, code writing, and manuscript preparation. Material preparation, data collection, and analysis were performed by NB. The first draft of the manuscript was written by NB and LS, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Università degli Studi di Udine within the CRUI-CARE Agreement. This work was supported by The General Directorate for Scientific Research and Technological Development (DGRSDT) of Algeria, to whom we are grateful.

This research has been partially developed within the Laboratory for Big-Data, IoT, Cyber Security (LABIC) funded by Friuli Venezia Giulia region (Italy), and the Laboratory for Artificial Intelligence for Human-Robot Collaboration (AI4HRC) funded by Fondazione Friuli (Italy).

Data availability Not applicable

Declarations

Conflict of interest Not applicable

Ethics approval and consent to participate Not applicable

Consent for publication Not applicable

Materials availability Not applicable

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit

line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Angeles, J.: Fundamentals of robotic mechanical systems: theory, methods, and algorithms. Springer, New York (2003)
- Abu-Dakka, F.J., Assad, I.F., Alkhdour, R.M., Abderahim, M.: Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots. *Int. J. Adv. Manufact. Technol.* **89**, 389–406 (2017)
- Zhang, J., Meng, Q., Feng, X., Shen, H.: A 6-DOF robot-time optimal trajectory planning based on an improved genetic algorithm. *Robot. Biomimetics* **5**, 1–7 (2018)
- Piazzzi, A., Visioli, A.: Global minimum-jerk trajectory planning of robot manipulators. *IEEE Trans. Industr. Electron.* **47**(1), 140–149 (2000)
- Gasparetto, A., Lanzutti, A., Vidoni, R., Zanotto, V.: Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning. *Robot. Comput. Integrated Manufact.* **28**(2), 164–181 (2012)
- Liu, X., Shen, Z.: Rapid smooth entry trajectory planning for high lift/drag hypersonic glide vehicles. *J. Optim. Theory Appl.* **168**, 917–943 (2016)
- Boryga, M., Graboś, A.: Planning of manipulator motion trajectory with higher-degree polynomials use. *Mech. Mach. Theory* **44**(7), 1400–1419 (2009)
- Mohamed, K., Elgamal, H., Elsharkawy, A.: Dynamic analysis with optimum trajectory planning of multiple degree-of-freedom surgical micro-robot. *Alex. Eng. J.* **57**(4), 4103–4112 (2018)
- Bureerat, S., Pholdee, N., Radpukdee, T., Jaroenapibal, P.: Self-adaptive MRPBIL-DE for 6D robot multiobjective trajectory planning. *Expert Syst. Appl.* **136**, 133–144 (2019)
- Gasparetto, A., Zanotto, V.: A technique for time-jerk optimal planning of robot trajectories. *Robot. Comput. Integrated Manufact.* **24**(3), 415–426 (2008)
- Gasparetto, A., Zanotto, V.: Optimal trajectory planning for industrial robots. *Adv. Eng. Softw.* **41**(4), 548–556 (2010)
- Nadir, B., Mohammed, O.: Optimization the trajectories of robot manipulators along specified task. *Int. J. Intell. Eng. Syst.* **11**(1), 11–19 (2018)
- Bendali, N., Ouali, M., Ghellal, K.: Optimal trajectory planning under kino-dynamics constraints for a 6-DOF PUMA 560. In: Proceedings of the The International Conference on Engineering & MIS 2015, pp. 1–4 (2015)
- Nadir, B., Mohammed, O., Abdessalam, K.: 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.* **3**(3), 184 (2014)
- Choi, Y.-K., Park, J.-H., Kim, H.-S., Kim, J.H.: Optimal trajectory planning and sliding mode control for robots using evolution strategy. *Robotica* **18**(4), 423–428 (2000)
- Constantinescu, D., Croft, E.A.: Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *J. Robot. Syst.* **17**(5), 233–249 (2000)
- Gallant, A., Gosselin, C.: Extending the capabilities of robotic manipulators using trajectory optimization. *Mech. Mach. Theory* **121**, 502–514 (2018)
- Saravanan, R., Ramabalan, S., Balamurugan, C.: Evolutionary optimal trajectory planning for industrial robot with payload constraints. *Int. J. Adv. Manufact. Technol.* **38**, 1213–1226 (2008)
- Saramago, S., Steffen, V.: Trajectory modeling of robot manipulators in the presence of obstacles. *J. Optim. Theory Appl.* **110**, 17–34 (2001)
- Visioli, A.: Trajectory planning of robot manipulators by using algebraic and trigonometric splines. *Robotica* **18**(6), 611–631 (2000)
- Wang, C.-H., Horng, J.-G.: Constrained minimum-time path planning for robot manipulators via virtual knots of the cubic B-spline functions. *IEEE Trans. Autom. Control* **35**(5), 573–577 (1990)
- Zhang, X., Shi, G.: Multi-objective optimal trajectory planning for manipulators in the presence of obstacles. *Robotica* **40**(4), 888–906 (2022)
- Gasparetto, A., Zanotto, V.: A new method for smooth trajectory planning of robot manipulators. *Mech. Mach. Theory* **42**(4), 455–471 (2007)
- Thompson, S.E., Patel, R.V.: Formulation of joint trajectories for industrial robots using B-splines. *IEEE Trans. Industr. Electron.* **2**, 192–199 (1987)
- Zanotto, V., Gasparetto, A., Lanzutti, A., Boscarriol, P., Vidoni, R.: Experimental validation of minimum time-jerk algorithms for industrial robots. *J. Intell. Robot. Syst.* **64**, 197–219 (2011)
- Wang, F., Wu, Z., Bao, T.: Time-jerk optimal trajectory planning of industrial robots based on a hybrid WOA-GA algorithm. *Processes* **10**(5), 1014 (2022)
- Huang, J., Hu, P., Wu, K., Zeng, M.: Optimal time-jerk trajectory planning for industrial robots. *Mech. Mach. Theory* **121**, 530–544 (2018)
- Zhu, S.W.: Time-optimal and jerk-continuous trajectory planning algorithm for manipulators. *J. Mechan. Eng.* **46**(3), 47–52 (2010)
- Zhou, Y., Han, G., Wei, Z., Huang, Z., Chen, X.: A time-optimal continuous jerk trajectory planning algorithm for manipulators. *Appl. Sci.* **13**(20), 11479 (2023)
- Xu, Z., Wei, S., Wang, N., Zhang, X.: Trajectory planning with Bézier curve in Cartesian space for industrial gluing robot. In: Intelligent Robotics and Applications: 7th International Conference, ICIRA 2014, Guangzhou, China, December 17–20, 2014, Proceedings, Part II 7, pp. 146–154 (2014). Springer
- Wang, C., Wang, X., Zheng, H.: Trajectory planning for a welding robot based on the Bézier curve. *Int. J. Control Autom.* **8**(2), 351–362 (2015)
- Zha, X.F.: Optimal pose trajectory planning for robot manipulators. *Mech. Mach. Theory* **37**(10), 1063–1086 (2002)
- Liu, Q., Wang, H.: Using Bézier Curve to Plan the Motion of the Robot Manipulator Based on Quaternion. *Journal of Information & Computational Science* **11**(7), 2299–2307

34. Lee, A.Y., Choi, Y.: Smooth trajectory planning methods using physical limits. In: Proc. Inst. Mech. Eng. C J. Mech. Eng. Sci. **229**(12), 2127–2143 (2015)
35. Li, H.: A jerk-constrained asymmetric motion profile for high-speed motion stages to reduce residual vibration. Int. J. Comput. Appl. Technol. **53**(2), 149–156 (2016)
36. Fang, Y., Hu, J., Liu, W., Shao, Q., Qi, J., Peng, Y.: Smooth and time-optimal S-curve trajectory planning for automated robots and machines. Mech. Mach. Theory **137**, 127–153 (2019)
37. Lambrechts, P., Boerlage, M., Steinbuch, M.: Trajectory planning and feedforward design for electromechanical motion systems. Control. Eng. Pract. **13**(2), 145–157 (2005)
38. Wu, Z., Chen, J., Bao, T., Wang, J., Zhang, L., Xu, F.: A novel point-to-point trajectory planning algorithm for industrial robots based on a locally asymmetrical jerk motion profile. Processes **10**(4), 728 (2022)
39. Perumaal, S., Jawahar, N.: Synchronized trigonometric S-curve trajectory for jerk-bounded time-optimal pick and place operation. Int. J. Robot. Autom. **27**(4), 385 (2012)
40. Liu, H., Lai, X., Wu, W.: Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. Robot. Comput. Integrated Manuf. **29**(2), 309–317 (2013)
41. Kucuk, S.: Optimal trajectory generation algorithm for serial and parallel manipulators. Robot. Comput. Integrated Manuf. **48**, 219–232 (2017)
42. Cook, C., Ho, C.: The application of spline functions to trajectory generation for computer-controlled manipulators. In: Computing Techniques for Robots, pp. 101–110 (1984). Springer
43. Petrinc, K., Kovacic, Z.: Trajectory planning algorithm based on the continuity of jerk. In: Mediterranean Conference on Control & Automation, pp. 1–5 (2007). IEEE
44. Boscariol, P., Gasparetto, A., Vidoni, R.: Jerk-Continuous Trajectories for Cyclic Tasks. Int. Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. Volume 4: 36th Mechanisms and Robotics Conference, Parts A and B, pp. 1277–1286 (2012)
45. Mirinejad, H., Inanc, T., Zurada, J.M.: radial basis function interpolation and Galerkin projection for direct trajectory optimization and costate estimation. IEEE/CAA J. Automatica Sinica **8**(8), 1380–1388 (2021)
46. Mirinejad, H.: A radial basis function method for solving optimal control problems. Electronic Theses and Dissertations, University of Louisville (2016)
47. Mirinejad, H., Inanc, T.: An RBF collocation method for solving optimal control problems. Robot. Auton. Syst. **87**, 219–225(2017)
48. Chettibi, T.: Smooth point-to-point trajectory planning for robot manipulators by using radial basis functions. Robotica **37**(3), 539–559 (2019)
49. Alipanah, A.: Multiquadric radial basis function method for the solution of brachistochrone problem. Rom. J. Math. Comput. Sci **6**, 126–133 (2016)
50. Nadir, B., Mohammed, O., Minh-Tuan, N., Abderrezak, S.: Optimal trajectory generation method to find a smooth robot joint trajectory based on multiquadric radial basis functions. Int. J. Adv. Manufact. Technol. **120**(1), 297–312 (2022)
51. Gu, J., Jung, J.-H.: Adaptive Gaussian radial basis function methods for initial value problems: Construction and comparison with adaptive multiquadric radial basis function methods. J. Comput. Appl. Math. **381**, 113036 (2021)
52. Hardy, R.L.: Multiquadric equations of topography and other irregular surfaces. J. Geophys. Res. **76**(8), 1905–1915 (1971)
53. Hardy, R.L.: Theory and applications of the multiquadric-biharmonic method 20 years of discovery 1968–1988. Comput. Math. Appl. **19**(8–9), 163–208 (1990)
54. Franke, R.: Scattered data interpolation: tests of some methods. Math. Comput. **38**(157), 181–200 (1982)
55. Press, W.H.: Numerical Recipes 3rd Edition: The Art of Scientific Computing, (2007). Cambridge University Press
56. Wang, L.: Radial basis functions methods for boundary value problems: Performance comparison. Eng. Anal. Boundary Elem. **84**, 191–205 (2017)
57. Rad, J.A., Kazem, S., Parand, K.: Radial basis functions approach on optimal control problems: a numerical investigation. J. Vib. Control **20**(9), 1394–1416 (2014)
58. Sundaram, A.: Combined heat and power economic emission dispatch using hybrid NSGA II-MOPSO algorithm incorporating an effective constraint handling mechanism. IEEE Access **8**, 13748–13768 (2020)
59. Sundaram, A., Erdogmus, P.: Solution of combined economic emission dispatch problem with valve-point effect using hybrid NSGA II-MOPSO. Particle Swarm Optimization with Applications **78** (2017)
60. Simon, D., Isik, C.: A trigonometric trajectory generator for robotic arms. Int. J. Control **57**(3), 505–517 (1993)
61. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)

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