Optimal time-jerk trajectory planning of a series-parallel robotic manipulator using novel generalized hyperbolic radial basis functions

Deva Karthik Lakshman Dasu[1*], Anirudh R[1] and Manu V Unnithan[1]

[1*]Aerospace Mechanisms Group, Vikram Sarabhai Space Center, Veli, Thiruvananthapuram, 695022, Kerala, India.

*Corresponding author(s). E-mail(s): karthiklakshmandasu@gmail.com;
Contributing authors: iiauthor@gmail.com; iiiauthor@gmail.com;

Note: This article is under internal review and has not yet been peer-reviewed.

1 Introduction

In recent years, series-parallel hybrid robotic manipulators (RM) have gained huge attention in the robotics community due to their edge over serial and parallel RMs for complex tasks. For example, palletizing tasks in industries demand features like long-range manipulation and higher load-carrying capacity within the optimal size and power constraints. The features are not found combined in conventional serial and parallel RMs. However, a hybrid series-parallel palletizing RM offers a better solution than the serial and parallel RMs, satisfying the design constraints [1]. The reason is due to its architecture; similar to the serial RMs, it consists of a base, a lower arm, an upper arm, and an end effector, which helps in long-range manipulation. Nevertheless, the major difference is that the base houses all the actuators, and the upper arm is actuated by a closed-loop parallel four-bar linkage from the base, which aids in high load-carrying capacity. Furthermore, the watt's linkage connecting the base to the end-effector forms two additional closed-loop kinematic chains that make the orientation of the end-effector constant throughout the motion. This results in less inertia in the upper body, leading to better control and accuracy over the tasks, making it

suitable for palletizing, robotic welding, spraying, additive manufacturing, etc [2–4]. The RM can also generate reactionless trajectories owing to its architecture [5], which is highly desirable for space applications like in-space assembly, manufacturing, on-orbit servicing, etc. Due to its simple and unique structure, the same RM configuration has been chosen for the "robotic 3D printing in space" experiment at VSSC.

The biggest challenge lies in planning the 3D printing task. Irrespective of the configuration, the tasks performed by the RM fall into two categories: Continuous Tasks (CTs) and Pick-and-Place Tasks (PPTs). In PPTs, the path of the end effector (EE) is free but has to satisfy the given initial and final positions by avoiding obstacles (if any), e.g., palletizing tasks. In CTs, the EE path is continuous and smooth, specified by a finite set of points (knots), e.g., 3D Printing, welding, and spraying. For a given task, it is necessary to optimize the trajectory and operation time of the RM to attain maximum productivity. The process is called the Trajectory Planning (TP) of the RM. The TP finds an interpolation between RM spatial movements and time, satisfying path constraints. It is the same for all types of tasks and is performed in either the End-Effector Space (ES) or Joint Space (JS) of the RM. TP in ES is easy to envisage and interpret since most tasks are defined in ES. However, singularities may arise because of inverse kinematics (IK) computations during trajectory propagation. For TP in JS, the task in ES is transformed to JS, and TP is performed for joint values in JS. Usually, TP in JS is preferred because of fewer IK computations and the ability to impose kinematic constraints (joint velocity, acceleration, and jerk) directly.

The major challenge in TP lies in determining the time profile for the RM spatial movements. The operation time should be as minimal as possible to minimize power consumption, particularly for space and industrial applications. Early research considered the TP as part of the optimal control problem and primarily focused on minimizing the total time satisfying dynamic and specified path constraints [6–9]. Later, Lin et al. [10] divided the optimal control problem into optimal trajectory planning and trajectory tracking and focused on the former part. The author formulated the TP problem as a Nonlinear Programming Problem (NLP) with minimizing time as the objective function, subjected to velocity, acceleration, and jerk constraints. Furthermore, the author validated the approach on the PUMA 560 RM joint trajectories. Many researchers have proposed various algorithms to obtain near-minimum trajectories by solving the NLP [11, 12]. With the increase in the progress of evolutionary algorithms in the late 20^{th} century, researchers introduced genetic-based approaches for minimum-time TP in which the total time is the cost function and the time step between the knots are the design variables [13, 14].

However, the decrease in total time causes rapid vibration and tear of the joints which takes a toll on the accuracy of the tasks, especially, CTs. The rapid motion of the RM is quantified by the third-order derivative of the position also known as, jerk. Early researchers identified the problem and proposed a minimum jerk path generation problem with two cost functions, absolute maximum jerk [15, 16], and integral of jerk square [17, 18]. Lin et al. [19, 20] proposed a methodology based on K-means clustering and particle swarm optimization for solving the minimum-jerk TP using cubic splines. The author formulated the minimum-jerk TP as a non-linear programming problem (NLP) with a measure of a maximum jerk as the objective function

subject to fixed time constraints. Most researchers fixed the total time while minimizing the jerk because as the jerk minimizes, the total time increases for smooth joint movements. Although the health of the RM is maintained, it affects the productivity and power consumption of the RM. The jerk and total time are the two contradicting objectives that one needs to find an optimal trade-off between them while planning the trajectory. Minimal-time TP with no rapid movements has been a research problem in robotic manipulators since their inception.

Researchers have proposed various methods to determine an optimal trade-off between the RM total time and joint rates. Cao et al. [21] proposed a two-level approach for generating time-optimal smooth trajectories using cubic splines. The trajectory is optimized for minimum jerk in the first level. Then the optimal time intervals are scaled down to obtain the minimum total time that satisfies the kinematic constraints. Gasparetto et al. [22–25] converted the multi-objective optimization problem into a single objective optimization problem and solved for optimal TP. The author used the weighted sum of total time and jerk measure as the objective function for the TP using B-splines and validated the algorithm using experiments. Later, many researchers followed the same approach for optimal time-jerk TP and solved the optimization problem using different evolutionary and hybrid methods [26]. With the development of heuristic multi-objective optimization algorithms, like NSGA-II [27], researchers formulated the time-optimal TP problem as an actual multi-objective optimization problem with total time and jerk measure as the objectives. The result would be a set of optimal solutions, also known as optimal Pareto front and one can choose the solution based on time-jerk trade-off analysis [28–32]. Wang et al. [33] proposed an Improved Elitist NSGA-II (INSGA-II) algorithm to solve the multi-objective optimal TP problem, the author used total time and integral of absolute torque fluctuations as the objectives. Furthermore, researchers proposed energy [34], time-energy [35], and time-energy-jerk [36] based TP methods with a concentration in minimizing energy consumption.

The trajectories of RM are generally made using smooth interpolation splines to satisfy the velocity, acceleration, and jerk constraints. Cubic splines (CS) are the most used algebraic splines for the TP of RMs. This is because the CS can limit the jerk and produce the lowest jerk peaks, even though the jerk profile is discontinuous [37]. Many researchers also used higher-order algebraic splines for smooth and continuous jerk profiles, these include quintic splines [22], B-splines [23, 30], trigonometric splines [38], Beizer curves [39], S-curves [31], neural networks [18], etc. Some researchers proposed hybrid methods by combining different interpolation methods. For example, ref. [37] used a seventh-order polynomial to generate zero jerk at initial and final knots for the trajectory obtained using cubic splines. Ref. [33] combined quintic polynomials with cubic Bezier curves to form a hybrid polynomial for TP.

In this study, we focus on the TP of the RMs using the Radial Basis Function (RBF) interpolation method. RBF interpolation is one of the applications of RBF Neural Networks (RBFNN), which are a form of artificial neural networks (ANN) used primarily for function approximation. The architecture of the RBFNN is simple and compact; it consists of an input layer, a hidden layer, and an output layer. The hidden layer uses non-linear RBFs, like Gaussian, multi-quadrics, etc. as the activation functions in the

neurons, which helps to approximate non-linear functions. Furthermore, some RBFs are infinitely smooth which is an essential requirement for planning smooth trajectories. Various types of RBFs have been introduced in the past for different applications, and one can design specific RBFs depending on the desired characteristics. Despite the advantages, very few reported the application of RBFs in the TP of the RMs. Chettibi et al. [28] used Gaussian RBFs for the TP of the RMs, the author validated the approach on the PUMA 560 RM. Nadir et al. [40] used Multi-quadric RBFs for the TP of the RMs, the author showed that the proposed method generated lesser jerk than the previous TP methods. Mirinejad et al. [41] used the RBF collocation method for solving optimal control problems in motion planning, the author further combined the RBF interpolation with Galerkin projection to obtain solutions for discontinuous and non-smooth problems in optimal control [42]. So far the researchers have used only the existing RBFs for the TP quoting the advantage of shape parameter in smoothness. There is no explanation for choosing a particular RBF. Furthermore, an attempt to develop an RBF to cater to the requirement of TP has not been made, despite the potential of the RBFs. In this paper, we introduce Generalized Hyperbolic Radial Basis Functions (GH-RBFs) which are designed for the minimum jerk requirement of the RM. There have been instances in the past using hyperbolic secant function as the RBF for classification and approximation purposes [43–45]. Nevertheless, in this study, we added an exponential parameter to the RBF in addition to the shape parameter which helps in planning smooth optimal trajectories. Moreover, the proposed RBF is validated against the PPTs and CTs using PUMA 560 RM. Finally, an optimal time-jerk trajectory is planned for an additive manufacturing task using the series-parallel hybrid robotic manipulator for the 3D printing in space experiment.

The remainder of this paper is structured as follows, section 2 defines the problem statement for the optimal time-jerk TP of the series-parallel hybrid RM. Section 3 presents the theoretical background, definition, and properties of the proposed GH-RBFs. Section 4 illustrates the comparative test cases for the robot trajectory planning using GH-RBFs. Section 5 shows the optimal time-jerk TP of series-parallel hybrid RM for an additive manufacturing task and, section 6 concludes the current study and offers an insight into the future scope.

2 Problem statement

In the current study, TP for the RM is formulated as a multi-objective optimization problem in JS. The time profile is discretized into finite time steps corresponding to each knot and the lengths of the time segments between the knots are considered as the design variables. Furthermore, the two conflicting objectives, the total time of operation and the total jerk measure are the objective functions. The total time is the sum of all time segments, i.e., design variables, and the jerk measure is quantified using the Jerk Index (JI), which is the sum of the square root of the integral of the square of jerk of each joint. Moreover, the RM trajectory must satisfy the maximum velocity,

acceleration, and jerk constraints. The TP problem is formulated as follows [22],

$$\begin{aligned} Min \quad & w_1 \sum_{i=1}^{N-1} T_i + w_2 \sum_{j=1}^{M} \sqrt{\frac{1}{T} \int_0^T \dddot{q}_j^{\,2} dt} \\ S.t., & \\ & |\dot{q}_i| \leq \dot{q}_{i max}, \qquad i = 1, \ ..., \ M \\ & |\ddot{q}_i| \leq \ddot{q}_{i max}, \qquad i = 1, \ ..., \ M \\ & |\dddot{q}_i| \leq \dddot{q}_{i\,max}, \qquad i = 1, \ ..., \ M \end{aligned} \tag{1}$$

Where $\{T_i\}$ are the design variables (time segments), $\{q_i, \ \dot{q}_i, \ \ddot{q}_i, \ \dddot{q}_i\}$ are the path, velocity, acceleration, and jerk of the RM joints. The first term in the objective function is the total time T and the second term is the Jerk Index (JI). The coefficients $\{w_1, \ w_2\}$ are the weight factors of the objective function, M denotes the total number of RM joints, and N represents the number of knots.

3 Generalized hyperbolic - radial basis functions (GH-RBF)

This section introduces the theoretical framework behind Generalized Hyperbolic Radial Basis Functions (GH-RBF) and their application in trajectory planning and approximation theory. RBF approximation is an advanced mesh-free method for interpolating scattered data. The RBFs are widely used in engineering and sciences for approximating differential operators, output functions, etc. An RBF can be defined as follows [46, 47],

Definition 1. *A function $\Phi : \mathbb{R}^s \to \mathbb{R}$ is said to be radial provided there exists a univariate function, $\phi : [0, \infty) \to \mathbb{R}$ such that*

$$\Phi(\mathbf{x}) = \phi(r)$$

where $r = \|\mathbf{x}\|$, and $\|.\|$ is some norm on $\mathbb{R}^s$, usually the euclidean norm.

The RBF value is constant for all the points with the same distance from the origin (or the center), which shows the radially symmetric nature of the RBF about the centers. The other important feature is that the RBF is independent of the dimension of space (depends only on the norm), making it suitable for high dimensional space applications [46]. The approximation or interpolation problem using RBF can be stated as follows[47],

Given the function values $f_1, \ f_2, \ ..., \ f_N \in \mathbb{R}$ for the distinct data sites $\mathbf{x_1}, \ \mathbf{x_2}, \ ..., \ \mathbf{x_N} \in \mathbb{R}^n$, we are interested in finding a continuous function $s : \mathbb{R}^n \to \mathbb{R}$ with

$$s(\mathbf{x_i}) = f_i, \qquad 1 \leq i \leq N \tag{2}$$

Hardy et al introduced the concept of RBF interpolation using multi-quadric functions to solve the above problem. In RBF interpolation, the approximant s is the linear

combination of the RBFs whose centers shifted from origin to c_i. It can be defined as follows,

$$s(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i \phi(\|\mathbf{x} - c_i\|), \qquad \mathbf{x} \in \mathbb{R}^s,\ s > 0 \tag{3}$$

Where N is the number of data sites, and λ_i are the real coefficients. Now there are $2N$ number of unknowns (c_i and λ_i) with N equations. By considering the data sites as the RBF centers ($c_i = x_i$), the number of unknowns is reduced to N. Substituting equation 3 in 2 gives the following system of linear equations,

$$\begin{bmatrix} \phi(\|\mathbf{x_1} - \mathbf{x_1}\|) & \phi(\|\mathbf{x_1} - \mathbf{x_2}\|) & \dots & \phi(\|\mathbf{x_1} - \mathbf{x_N}\|) \\ \phi(\|\mathbf{x_2} - \mathbf{x_1}\|) & \phi(\|\mathbf{x_2} - \mathbf{x_2}\|) & \dots & \phi(\|\mathbf{x_2} - \mathbf{x_N}\|) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \phi(\|\mathbf{x_N} - \mathbf{x_1}\|) & \phi(\|\mathbf{x_N} - \mathbf{x_2}\|) & \dots & \phi(\|\mathbf{x_N} - \mathbf{x_N}\|) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \dots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ \dots \\ f_N \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} \phi_1(0) & \phi_2(\mathbf{x_1}) & \dots & \phi_m(\mathbf{x_1}) \\ \phi_1(\mathbf{x_2}) & \phi_2(0) & \dots & \phi_m(\mathbf{x_2}) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \phi_1(\mathbf{x_m}) & \phi_2(\mathbf{x_m}) & \dots & \phi_m(0) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \dots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ \dots \\ f_N \end{bmatrix} \tag{5}$$

$$\mathbf{A}_{N\times N}\lambda_{N\times 1} = \mathbf{F}_{N\times 1} \tag{6}$$

Where the real coefficients $\lambda_{N\times 1}$ are obtained by inverting the interpolation matrix $\mathbf{A}_{N\times N}$. The invertibility of the interpolation matrix plays a major role in RBF interpolation.

3.1 Theoretical background

As discussed earlier, the crucial requirements for efficient and optimal trajectory planning are less joint acceleration and jerk. In TP using RBFs, the joint trajectory is expressed as RBF interpolation (as in equation (3)). For a given coefficient vector $\lambda_{N\times 1}$, the magnitude of the velocity, acceleration, and jerk depends on the higher derivatives of RBFs. The higher derivatives need not be radial and may have magnitudes greater than the RBF over the domain. The TP requires RBFs whose derivative magnitude is less than those across the domain. Many RBFs fit into this category and one such function is $e^{-\frac{r}{c}}$. Since the joint trajectory is univariate with time, the Euclidean norm in the RBF can be replaced with the absolute function.

$$\phi(x) = e^{-\left(\left|\frac{|x-x_i|}{c}\right|\right)},\ c \neq 0 \tag{7}$$

For $c > 1$, the magnitudes of the higher derivatives are lesser than the actual RBF values by a factor of c. Furthermore, as the order of derivative increases, the magnitude decreases over the domain. However, the higher derivatives are discontinuous

at the centers which is undesirable for the TP. We replace the absolute function in the exponent with its smooth equivalent $log\left[cosh\left(\left|\frac{x-x_i}{c}\right|\right)\right]$, which is widely used in machine learning log-cosh loss function [48]. The RBF in the equation (7) becomes,

$$\begin{aligned}\phi(x) &= e^{-log\left[cosh\left(\left|\frac{x-x_i}{c}\right|\right)\right]} \\ &= e^{log\left[cosh\left(\left|\frac{x-x_i}{c}\right|\right)\right]^{-1}} \\ &= \left[cosh\left(\left|\frac{x-x_i}{c}\right|\right)\right]^{-1}, \ c \neq 0\end{aligned} \tag{8}$$

We substitute the exponent in the equation with a real exponent n to define a more generalized function. Furthermore, since the hyperbolic cosine function is even, the absolute function can be dropped and expressed as,

$$\phi(x) = cosh^n\left(\frac{x-x_i}{c}\right), \ c, \ n \neq 0 \tag{9}$$

Moreover, we observed that the function definition in equation (9) is not complete, since the function is non-strictly positive definite for positive integer values of n (similar to Generalized Multi-Quadric RBFs (GMQ-RBF)). The positive definiteness of the function for all values of n is discussed in Theorems 3.1 and 3.3.

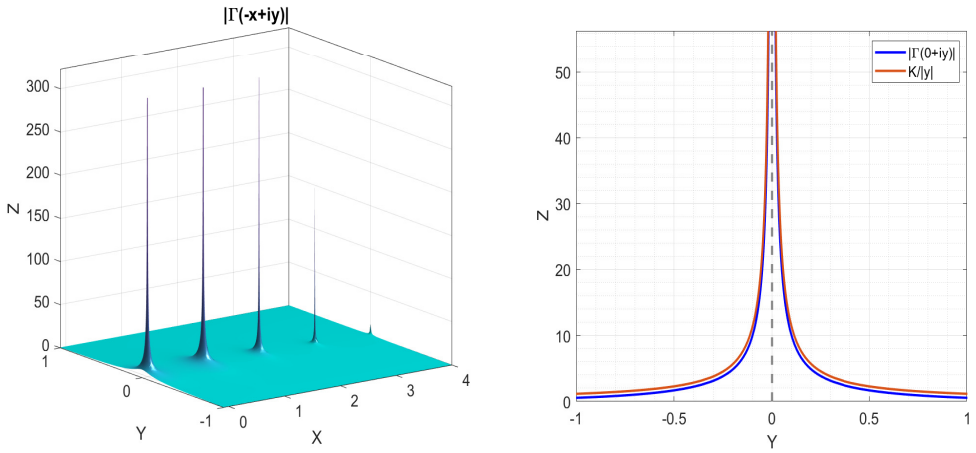

Fig. 1 Absolute gamma function $\Gamma(-x+iy)$ (left) and comparison of absolute gamma function and $K/|y|$ (right).

Theorem 3.1. *The function* $\Phi(x) = sech^{\beta}(\frac{x}{c})$, $x \in \mathbb{R}^1$ *with* $c > 0$ *and* $\beta > 0$ *is strictly positive definite.*

Proof. From the definition of Γ- function for $\beta > 0$ we see that

$$\Gamma(\beta) = \int_0^\infty t^{\beta-1} e^{-t} dt \tag{10}$$

Put $t = su$ with $s > 0$. we get

$$\Gamma(\beta) = s^\beta \int_0^\infty u^{\beta-1} e^{-su} du \tag{11}$$

By substituting $s = cosh(\frac{x}{c})$, we get

$$sech^\beta\left(\frac{x}{c}\right) = \frac{1}{\Gamma(\beta)} \int_0^\infty u^{\beta-1} e^{-ucosh(\frac{x}{c})} du \tag{12}$$

Inserting this into the Fourier Transform leads to

$$\begin{aligned} \hat{\Phi}(\omega) &= (2\pi)^{-\frac{1}{2}} \int_{\mathbb{R}^1} \Phi(x) e^{-ix\omega} dx \\ &= (2\pi)^{-\frac{1}{2}} \frac{1}{\Gamma(\beta)} \int_{\mathbb{R}^1} \int_0^\infty u^{\beta-1} e^{-ucosh(\frac{x}{c})} du e^{-ix\omega} dx \\ &= (2\pi)^{-\frac{1}{2}} \frac{1}{\Gamma(\beta)} \int_0^\infty u^{\beta-1} \left[\int_{\mathbb{R}^1} e^{-ucosh(\frac{x}{c})} e^{-ix\omega} dx \right] du \end{aligned} \tag{13}$$

We know that from the definition of *Modified Bessel Function of Third kind of the Imaginary Order* [49].

$$2K_{i\nu}(u) = \int_{-\infty}^\infty e^{-ucosh(x)-i\nu x} dx, \;\; Re(u) > 0 \tag{14}$$

By substituting $p = \frac{x}{c}$ and using equations (13) and (14), the fourier transform leads to

$$\hat{\Phi}(\omega) = (2\pi)^{-\frac{1}{2}} \frac{2c}{\Gamma(\beta)} \int_0^\infty u^{\beta-1} K_{ic\omega}(u) du \tag{15}$$

Here $\beta > 0$ and $Re(ic\omega) = 0$. We know that from integrals of modified bessel function of third kind of imaginary order [50].

$$\int_0^\infty u^{\beta-1} K_{ic\omega}(u) du = 2^{\beta-2} \Gamma\left(\frac{\beta + ic\omega}{2}\right) \Gamma\left(\frac{\beta - ic\omega}{2}\right) \tag{16}$$

Put $z = \frac{\beta+ic\omega}{2}$. We know that $\Gamma(z)\Gamma(\overline{z}) = |\Gamma(z)|^2$, the fourier transform becomes,

$$\hat{\Phi}(\omega) = (2\pi)^{-\frac{1}{2}} \frac{2^{\beta-1} c}{\Gamma(\beta)} \left| \Gamma\left(\frac{\beta}{2} + \frac{ic\omega}{2}\right) \right|^2 \tag{17}$$

For $\beta > 0$ and $\omega \in \mathbb{R}$ the Fourier transform is non-negative and non-vanishing. From Theorem 8.12 in [46], the function $\Phi(x) = sech^{\beta}(\frac{x}{c})$ is strictly positive definite. □

Lemma 3.2. *For $x \geq 0$ ($x \in \mathbb{R}$) and y (< 1) $\to 0$ the absolute gamma function has the following bound*

$$|\Gamma(-x+iy)| < \frac{K}{|y|}$$

Where $K = max\left\{\frac{1}{\Gamma(x)}\right\}$ *for* $x \geq 0$

Proof. We know that from the definition of Gamma function

$$\begin{aligned}\Gamma(z) &= \frac{1}{z}\prod_{n=1}^{\infty}\frac{\left(1+\frac{1}{n}\right)^{z}}{\left(1+\frac{z}{n}\right)} \\ &= \frac{1}{z}\prod_{n=1}^{\infty}\frac{\left(1+\frac{1}{n}\right)^{z}\left(1-\frac{z}{n}\right)}{\left(1-\frac{z^2}{n^2}\right)} \\ &= \frac{1}{z}\frac{\prod_{n=1}^{\infty}\left(1+\frac{1}{n}\right)^{z}\left(1-\frac{z}{n}\right)}{\prod_{n=1}^{\infty}\left(1-\frac{z^2}{n^2}\right)}\end{aligned} \tag{18}$$

We know from the definition of the sine function, $sin(\pi z) = \pi z\prod_{n=1}^{\infty}\left(1-\frac{z^2}{n^2}\right)$,

$$\Gamma(z) = \frac{\pi}{sin(\pi z)}\prod_{n=1}^{\infty}\left(1+\frac{1}{n}\right)^{z}\left(1-\frac{z}{n}\right) \tag{19}$$

Substituting $z = -x+iy$ in the equation leads to,

$$\Gamma(-x+iy) = \frac{\pi}{sin(-\pi x+i\pi y)}\prod_{n=1}^{\infty}\left(1+\frac{1}{n}\right)^{-x+iy}\left(1+\frac{x}{n}-\frac{iy}{n}\right) \tag{20}$$

Here as $y \to 0$, $\left|\left(1+\frac{x}{n}-\frac{iy}{n}\right)\right| \approx \left|\left(1+\frac{x}{n}\right)\right|$ and $\left|\left(1+\frac{1}{n}\right)^{-x+iy}\right| = \left|\left(1+\frac{1}{n}\right)^{-x}\right|$, the equation now becomes

$$|\Gamma(-x+iy)| \approx \left|\frac{\pi}{sin(-\pi x+i\pi y)}\right|\prod_{n=1}^{\infty}\left|\left(1+\frac{1}{n}\right)^{-x}\left(1+\frac{x}{n}\right)\right| \tag{21}$$

From the definition of $\left|\frac{1}{\Gamma(x+1)}\right|$, the equation leads to

$$|\Gamma(-x+iy)| \approx \left|\frac{\pi}{sin(-\pi x+i\pi y)}\right|\left|\frac{1}{\Gamma(x+1)}\right| \tag{22}$$

Now, using the equation (22) we see the following cases
Case (i) As $y \to 0$ and $\forall\ x \in \mathbb{N}_0$, where the reciprocal gamma function is less than or equal to one, $\frac{1}{\Gamma(x+1)} \leq 1$

$$\begin{aligned} |\Gamma(-x+iy)| &\leq \left|\frac{\pi}{sin(i\pi y)}\right| \\ &\leq \frac{1}{|y|} \end{aligned} \tag{23}$$

Case (ii) As $y \to 0$ and $\forall\ x \in (0,1)$, the reciprocal gamma function is greater than one, $\frac{1}{\Gamma(x+1)} > 1$
Here as $y \to 0$,

$$\begin{aligned} &sin(-\pi x + i\pi y) \approx -sin(\pi x) + icos(\pi x)\pi y \\ &|-sin(\pi x) + icos(\pi x)\pi y| \geq \pi|y| \\ &\left|\frac{1}{-sin(\pi x) + icos(\pi x)\pi y}\right| \leq \frac{1}{\pi|y|} \end{aligned} \tag{24}$$

Let $K = max\left\{\frac{1}{\Gamma(x+1)}\right\}$, and $\left|\frac{1}{\Gamma(x+1)}\right| \leq K,\ \forall\ x \in (0,\ 1)$

$$\begin{aligned} |\Gamma(-x+iy)| &\leq \left|\frac{K\pi}{sin(-\pi x + i\pi y)}\right| \\ &\leq \left|\frac{K\pi}{\pi y}\right| \\ &\leq \frac{K}{|y|} \end{aligned} \tag{25}$$

Case (iii) As $y \to 0$ and $\forall x \in \mathbb{R} \setminus \{\mathbb{N}_0 \cup\ (0,\ 1)\}$ the reciprocal gamma function $\frac{1}{\Gamma(x+1)} \leq 1$ and from the inequality (24)

$$\begin{aligned} |\Gamma(-x+iy)| &\leq \left|\frac{\pi}{\pi y}\right| \\ &\leq \frac{1}{|y|} \end{aligned} \tag{26}$$

From cases (i), (ii) and (iii) the absolute gamma function has the bound for $x\ \geq 0$ ($x \in \mathbb{R}$) and $y \to 0$ (also shown in figure 1)

$$|\Gamma(-x+iy)| \leq \frac{K}{|y|} \tag{27}$$

□

Theorem 3.3. *The function* $\Phi(x) = cosh^{\beta}(\frac{x}{c})$, $x \in \mathbb{R}^1$ *with* $c > 0$ *and* $\beta \in \mathbb{R} \setminus \mathbb{N}_0$ *possesses the generalized-fourier transform of order m.*

$$\hat{\Phi}(\omega) = (2\pi)^{\frac{-1}{2}} \frac{2^{-\beta-1}c}{\Gamma(-\beta)} \left| \Gamma\left(\frac{-\beta}{2} + \frac{ic\omega}{2}\right) \right|^2, \quad \omega \neq 0 \tag{28}$$

where $m = max(0, \lceil \beta \rceil)$

Proof. We prove the theorem the same way as Theorem 8.15 from Reference [46] using analytic continuation. Define $G = \{\lambda \in \mathbb{C} : Re(\lambda) < m\}$ and let us denote the right-hand side of the equation (28) by $\phi_\beta(\omega)$ and the function $cosh^{\lambda}(\frac{\omega}{c})$ by $\Phi_\lambda(\omega)$. Now we are going to show by analytic continuation that for all $\lambda \in G$,

$$\int_{\mathbb{R}} \Phi_\lambda(\omega)\hat{\gamma}(\omega)d\omega = \int_{\mathbb{R}} \phi_\lambda(\omega)\gamma(\omega)d\omega, \ \gamma \in S_{2m} \tag{29}$$

Note that the above equation holds for $\lambda < 0$ (from Theorem 3.1) and for $\lambda = 0\ ,1\ ,2\ ,...,\ m-1$ for $m > 0$ (since $\frac{1}{\Gamma(-\lambda)} = 0$). From Theorem 8.15 [46], it is now sufficient to show that the integrand in the right-hand side of equation (29) is bounded uniformly on a closed curve C in G.
Let us investigate the asymptotic characteristics of the integrand near the origin, say for $|\omega| << min\{\frac{1}{c},\ 1\}$. Let $x = Re(\lambda)$ and using the Lemma. 3.2 the integrand has the following behaviour,

$$|\phi_\lambda(\omega)\gamma(\omega)d\omega| \leq C_\gamma (2\pi)^{\frac{-1}{2}} \frac{2^{-x-1}}{|\Gamma(-\lambda)|} \frac{4K^2}{c} |\omega|^{2m-2} \tag{30}$$

Here C is compact and $\frac{1}{\Gamma(-\lambda)}$ is analytic, now for all $\lambda \in C$ and $|\omega| < min\{\frac{1}{c},\ 1\}$, the inequality can be written as,

$$|\phi_\lambda(\omega)\gamma(\omega)d\omega| \leq C_{\gamma,m,c,C} |\omega|^{2m-2} \tag{31}$$

Now for large arguments, $\omega \to \infty$ using inequality from the reference

$$|\phi_\lambda(\omega)\gamma(\omega)d\omega| \leq C_\gamma \frac{2^{-x-1}c}{|\Gamma(-\lambda)|} |\omega|^{-(x+1)/2} e^{-\pi c|\omega|/4} \tag{32}$$

Here $\gamma \in S$ is bounded and since C is compact the integrand can be bounded by,

$$|\phi_\lambda(\omega)\gamma(\omega)d\omega| \leq C_{\gamma,m,c,C} e^{-\pi c|\omega|/4} \tag{33}$$

Here both the inequalities are independent of λ, hence the proof is complete. □

Theorem 3.4. *The function* $\Phi(x) = (-1)^m cosh^{\beta}(\frac{x}{c})$, $x \in \mathbb{R}^1$ *with* $c > 0$, $\beta \in \mathbb{R} \setminus \mathbb{N}_0$ *and* $m = max(0, \lceil \beta \rceil)$ *is conditionally positive definite of order m.*

Proof. The theorem can be proved from Theorems. 3.1 and 3.3. □

For $\beta \in \mathbb{R} \setminus \mathbb{N}_0$, the interpolation matrix is non-singular and can be used for RBF interpolation. And the negative coefficient in the theorem. 3.4 appears with the weights and can be removed. With this, we define the Generalized Hyperbolic-Radial Basis Function (GH-RBF) for $x \in \mathbb{R}^1$,

$$\phi(x) = cosh^n\left(\frac{r}{c}\right), \ c > 0, \ n \in \mathbb{R} \setminus \mathbb{N}_0 \tag{34}$$

Where $r = |x - x_i|$ and x_i are the data sites.

3.2 Interpolation using GH-RBF

The GH-RBF interpolation is validated using the function $f(x) = e^{xcos(3\pi x)}$, $x \in [0,\ 1]$. The interval is divided into fifteen equally spaced points as data sites for interpolation. The weights are computed using the equation (6) for various values of n. The interpolation is then compared with GMQ and Gaussian RBFs. Fig. 2 shows the interpolation plots for all the RBFs.

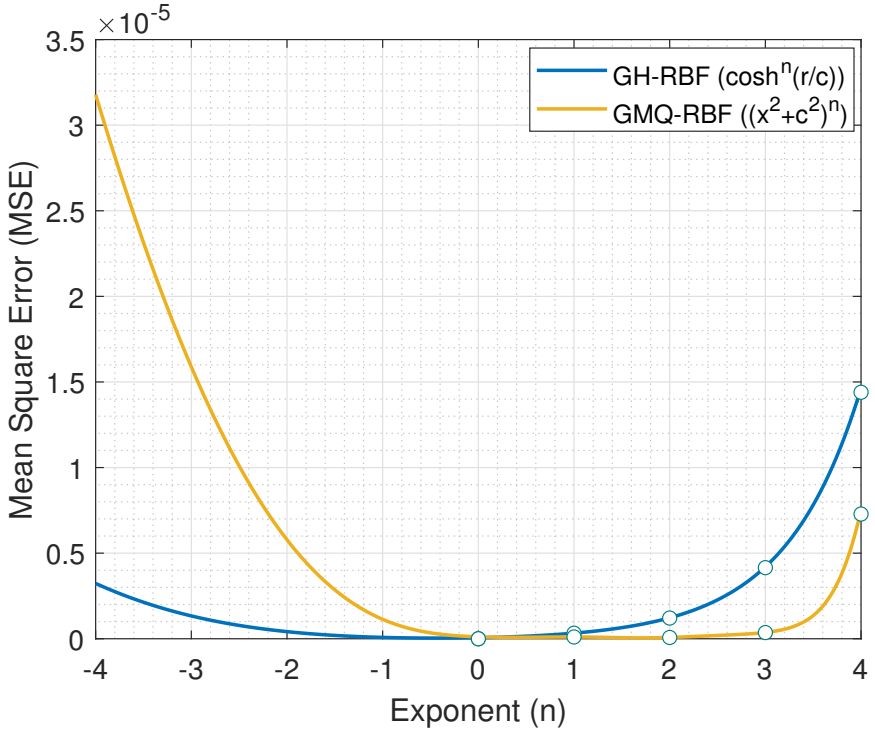

Fig. 2 Variation of Mean Square Error (MSE) with exponent (n)

Table 1 Mean Square Error (MSE) for positive integer exponents

Exponent(n)	GH-RBF	GMQ-RBF
1	10.5	3.3
2	6673.4	61.3
3	940.8	5.8
4	362.5	0.1

Note that from Fig. 2, both GH-RBF and GMQ-RBF are more accurate than Gaussian RBF. For $0 < n < 1$, the MSE of GH-RBF and GMQ-RBF are nearly comparable. For $n > 1$, the MSE of GH-RBF increased at a higher rate than that of GMQ-RBF, whereas the reverse is true for $n < 0$. Table 1 reports the MSE for the GH-RBF and GMQ-RBF at non-negative integers ($n = \{1, 2, 3, 4\}$) (shown as discontinuities in Fig. 2). The MSE values are several orders of magnitude higher than those shown in Fig. 2 because the RBFs are not strictly positive definite for positive integers.

3.3 Trajectory Planning using GH-RBF

The GH-RBF interpolation method is infinitely smooth, meaning all higher derivatives are continuous throughout the domain. This property is pivotal for planning optimal trajectories since the acceleration and jerk profiles are smooth and continuous. The derivatives of RBF interpolation are the linear combination of derivatives of RBFs. The derivatives of GH-RBF can be expressed as,

$$\frac{\partial \phi(r)}{\partial x} = \frac{n}{c} tanh\left(\frac{x - x_i}{c}\right) cosh^n\left(\frac{x - x_i}{c}\right) = \frac{n}{c} tanh\left(\frac{x - x_i}{c}\right) \phi(r) \tag{35}$$

$$\frac{\partial^2 \phi(r)}{\partial x^2} = \left[n + n(n-1) tanh^2\left(\frac{x - x_i}{c}\right)\right] \frac{\phi(r)}{c^2} \tag{36}$$

$$\frac{\partial^3 \phi(r)}{\partial x^3} = \left[n(n^2 - 3n + 2) tanh^3\left(\frac{x - x_i}{c}\right) + n(3n - 2) tanh\left(\frac{x - x_i}{c}\right)\right] \frac{\phi(r)}{c^3} \tag{37}$$

One of the major reason for introducing the exponent is that, for $n \in (-0.68, 1] \setminus \{0\}$, the magnitude of the first term in the equations (35), (36), and (37) are less than one for all values of x. The exponent parameter will further reduce the magnitude of higher derivatives, in addition to the shape parameter. Since TP requires lesser acceleration and jerk profiles, the same intervals are considered for n and c for the remainder of the study. The other essential requirement for TP is to satisfy the velocity and acceleration boundary conditions.

$$v_1 = \frac{\partial f}{\partial x}\Big|_{x=x_1}, \qquad a_1 = \frac{\partial^2 f}{\partial x^2}\Big|_{x=x_1} \tag{38}$$

$$v_n = \frac{\partial f}{\partial x}\Big|_{x=x_n}, \qquad a_n = \frac{\partial^2 f}{\partial x^2}\Big|_{x=x_n} \tag{39}$$

In TP using RBF interpolation, time is the independent variable, and the time points are the RBF data sites, which are arranged in sequential ascending order $\{x_1 < x_2 < ... < x_n\}$. One way to insert the boundary conditions is to create two virtual points at each end of the time profile as additional knots. Now, the equation (3) becomes,

$$f(x) = \sum_{i=1}^{N+4} \lambda_i \phi(\|x - x_i\|) \tag{40}$$

The interpolation matrix A becomes,

$$A_{N+4\times N+4} = \begin{bmatrix} \phi_1(0) & \phi_2(x_1) & \dots & \phi_N(x_1) & \phi_{N+1}(x_1) & \phi_{N+2}(x_1) & \phi_{N+3}(x_1) & \phi_{N+4}(x_1) \\ \phi_1(x_2) & \phi_2(0) & \dots & \phi_N(x_2) & \phi_{N+1}(x_2) & \phi_{N+2}(x_2) & \phi_{N+3}(x_2) & \phi_{N+4}(x_2) \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ \phi_1(x_N) & \phi_2(x_N) & \dots & \phi_N(0) & \phi_{N+1}(x_N) & \phi_{N+2}(x_N) & \phi_{N+3}(x_N) & \phi_{N+4}(x_N) \\ \phi_1'(x_{N+1}) & \phi_2'(x_{N+1}) & \dots & \phi_N'(x_{N+1}) & \phi_{N+1}'(x_{N+1}) & \phi_{N+2}'(x_{N+1}) & \phi_{N+3}'(x_{N+1}) & \phi_{N+4}'(x_{N+1}) \\ \phi_1'(x_{N+2}) & \phi_2'(x_{N+2}) & \dots & \phi_N'(x_{N+2}) & \phi_{N+1}'(x_{N+2}) & \phi_{N+2}'(x_{N+2}) & \phi_{N+3}'(x_{N+2}) & \phi_{N+4}'(x_{N+2}) \\ \phi_1''(x_{N+3}) & \phi_2''(x_{N+3}) & \dots & \phi_N''(x_{N+3}) & \phi_{N+1}''(x_{N+3}) & \phi_{N+2}''(x_{N+3}) & \phi_{N+3}''(x_{N+3}) & \phi_{N+4}''(x_{N+3}) \\ \phi_1''(x_{N+4}) & \phi_2''(x_{N+4}) & \dots & \phi_N''(x_{N+4}) & \phi_{N+1}''(x_{N+4}) & \phi_{N+2}''(x_{N+4}) & \phi_{N+3}''(x_{N+4}) & \phi_{N+4}''(x_{N+4}) \end{bmatrix} \tag{41}$$

The derivatives in the last four rows of the interpolation matrix are computed using the equation (35) and (36). Without loss of generality, the two virtual points are inserted near the endpoints as [28, 40],

$$\begin{aligned} x_{N+1} &= x_1 + 0.001(x_2 - x_1); \quad x_{N+2} = x_N - 0.001(x_N - x_{N-1}); \\ x_{N+3} &= x_1 + 0.002(x_2 - x_1); \quad x_{N+4} = x_N - 0.002(x_N - x_{N-1}) \end{aligned} \tag{42}$$

Accordingly, the coefficient vector $\{\lambda\}$ changes by,

$$\lambda_{N+4\times 1} = [\lambda_1,\ \lambda_2,\ ..., \lambda_N,\ \lambda_{N+1},\ \lambda_{N+2},\ \lambda_{N+3},\ \lambda_{N+4}]^T \tag{43}$$

and the vector $\{F\}$ changes by,

$$F_{N+4\times 1} = [f_1,\ f_2,\ ..., f_N,\ v_1,\ v_N,\ a_1,\ a_N]^T \tag{44}$$

Similarly, one can add the jerk boundary condition by inserting two more virtual points at the ends of the tie profile.

4 Comparative Test Cases for Robot Trajectory Planning

This section provides a comparative study of TP using GH-RBF with recent works using RBFs.

4.1 Pick and Place Tasks (PPT)

As discussed earlier, for the PPT, we are interested in finding a trajectory between two points q_i and q_f following null velocity and acceleration boundary conditions. For the first numerical test case, we study the trajectory of a single link with $q_i = 0$, $q_f = \pi$ and $t_f = 1.5s$. The trajectory is computed for the GH-RBF using the formulation discussed in sec. 3.3 with $v_1 = v_N = a_1 = a_N = 0$. The trajectory is then compared with GMQ-RBF and Gaussian RBFs. We assume the shape parameter $c = t_f$ is same for all the RBFs [28, 40]. Figure 3 shows the trajectory for the PPT and Table 2 reports the maximum values of the kinematic parameters of the trajectory.

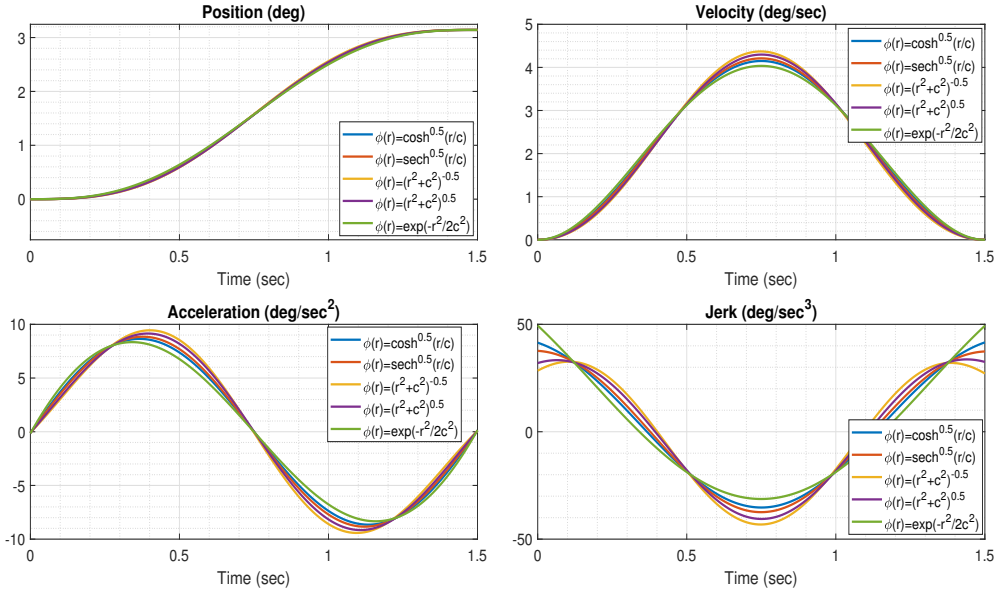

Fig. 3 Trajectory for Pick and Place Task

From Fig. 3, the position time history is nearly identical for all the RBFs. Furthermore, the profile is smooth at the endpoints without overshooting, following the null velocity and acceleration boundary conditions. The Gaussian RBF has better velocity and acceleration profiles with minimal extreme values than other RBFs. However, the maximum jerk for the Gaussian RBF is higher than those of the other RBFs. The Inverse GH-RBF (or $sech^n(r/c)$) reported a lower jerk value, which is 7.26% less than

Table 2 Maximum values of velocity, acceleration, and jerk for PPT

RBF	v_{max} (deg/sec)	a_{max} (deg/sec^2)	j_{max} (deg/sec^3)
$\phi(r) = cosh^{0.5}(\frac{r}{c})$	4.1490	8.6526	41.4583
$\phi(r) = sech^{0.5}(\frac{r}{c})$	4.2099	8.8481	**37.6593**
$\phi(r) = (r^2 + c^2)^{-0.5}$	4.3705	9.4459	43.1905
$\phi(r) = (r^2 + c^2)^{0.5}$	4.2977	9.1566	40.6071
$\phi(r) = exp(-\frac{r^2}{2c^2})$	**4.0355**	**8.3432**	49.4537

the GMQ-RBFs with the same exponential coefficient and shape parameter. In addition, the maximum velocity and acceleration of GH-RBFs are smaller than those of GMQ-RBFs.

4.2 Continuous Tasks (CT)

Continuous Tasks (CT) present two major cases for TP: minimum-time continuous tasks and minimum-time-jerk continuous tasks. For the former, the weight factor (w_2) in the objective function in equation (1) is zero since the objective minimizes only the total time. In the latter, both the weight factors are non-zero since there are two objectives. In the current study, PUMA 560, an industrial 6-axis robotic arm, is chosen for both cases.

4.2.1 Minimum time continuous tasks

In the first case, there are a total of six joints whose trajectories need to pass through eight different knots for each joint. The goal is to minimize the time required to traverse the knots (reported in Table 3) while satisfying the kinematic constraints for the joints (shown in Table 4). In addition to the time steps (h_i), the shape parameters (c_i) and exponents (n_i) are considered design variables for the optimization problem.

Table 3 Knots for Minimum time Trajectory Planning in JS

Knot	Joint (deg.)					
	J1	J2	J3	J4	J5	J6
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

To compare the result with earlier studies, the boundary conditions for acceleration and velocity are set to zero while the jerk is set as free. The optimal design variables of the TP are listed in table 5. Figure 4 shows the optimal position, velocity, acceleration, and jerk time histories obtained using GH-RBF. The total time for

Table 4 Kinematic Constraints for Minimum time Trajectory Planning in JS

Constraints	Joint (deg.)					
	J1	J2	J3	J4	J5	J6
Velocity (deg/sec)	100	95	100	150	130	110
Acceleration (deg/sec^2)	45	40	75	70	90	80
Jerk (deg/sec^3)	60	60	55	70	75	70

the trajectory obtained using GH-RBFs is 17.2081 s. The total time obtained using MQ-RBF [40] is 17.2347 s, and Gaussian RBF [28] is 17.7107 s. The position, velocity, acceleration, and jerk time histories for GH-RBF are continuous and equivalent to those obtained by Bendali [40] and Chettibi [28]. This shows that the GH-RBFs can produce minimal time trajectories satisfying kinematic constraints, which is essential for industrial applications.

Table 5 Optimal design variables for Minimum time Trajectory Planning

Design Variable	1	2	3	4	5	6	7
Time steps (h_i) (sec)	3.3005	1.1861	1.8410	2.3542	2.4567	2.9704	3.0991
Shape parameter (c_i)	6.7958	9.4999	1.9050	1.1036	5.5746	1.3931	-
Exponent (n_i)	-0.2569	0.3551	-0.6465	0.1520	0.2012	0.6695	-

4.2.2 Minimum time-jerk continuous tasks

The objective in the second case is to plan a smooth trajectory for the 6-DoF PUMA 560 robot manipulator. The test case includes four knots for each of the six joints that have to satisfy the given kinematic constraints. The overall objective is to minimize the jerk index while reducing the total time required to generate smooth trajectories. Table 6 and 7 report the knots and kinematic constraints for the TP.

Table 6 Knots for Minimum time-jerk Trajectory Planning in JS

Knot	Joint (deg.)					
	J1	J2	J3	J4	J5	J6
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

To compare the results obtained with the previous research [40], the weight factors (w_1) and (w_2) in the equation (1) are modified to match the total time ($t_f = 9.1\ s$). Furthermore, the velocity and acceleration profiles are given null boundary conditions while the jerk profile is given free boundary conditions. Table 8 and figure 5 show the

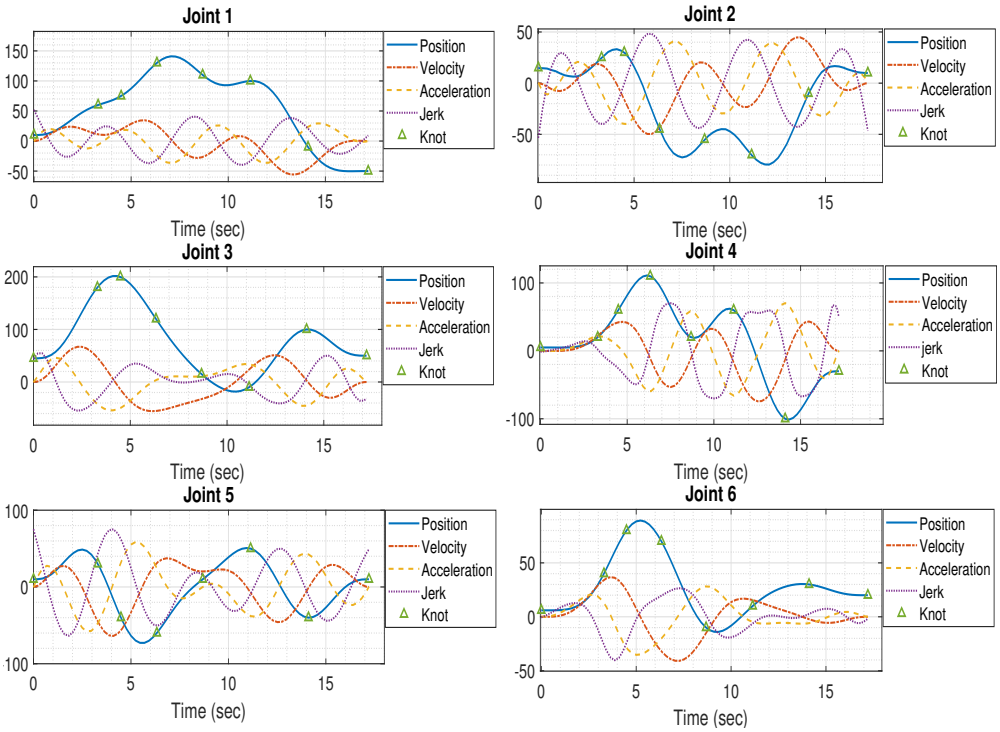

Fig. 4 Trajectory for minimum time CT

Table 7 Kinematic Constraints for Minimum time-jerk Trajectory Planning in JS

Constraints	Joint (deg.)					
	J1	J2	J3	J4	J5	J6
Velocity (deg/sec)	100	95	100	150	130	110
Acceleration (deg/sec^2)	60	60	75	70	90	80
Jerk (deg/sec^3)	60	66	85	70	75	70

optimal design variables and the optimal trajectory of the joints. The time histories of velocity, acceleration, and jerk are continuous and satisfy the boundary conditions and kinematic constraints. Table 9 shows the mean kinematic values of the velocity, acceleration, and jerk obtained using the GH-RBF. Comparing with the previous studies [40], the mean jerks attained by GH-RBF for all the joints are lower than those obtained by the MQ-RBF by 4.66%, 4.67%, 2.82%, 12.79%, 10.49%, and 4.99% respectively. It is evident that the GH-RBFs illustrate better performance compared to the previous methods, in terms of jerk, which is crucial for planning CTs.

Table 8 Optimal design variables for minimum time-jerk Trajectory Planning

Design Variable	1	2	3	4	5	6
Time steps (h_i) (sec)	3.3551	2.5642	3.1807	-	-	-
Shape parameter (c_i)	7.4163	1.4480	6.0656	9.9038	1.4434	8.5024
Exponent (n_i)	0.1240	-0.1009	0.9090	0.2435	-0.1662	-0.3101

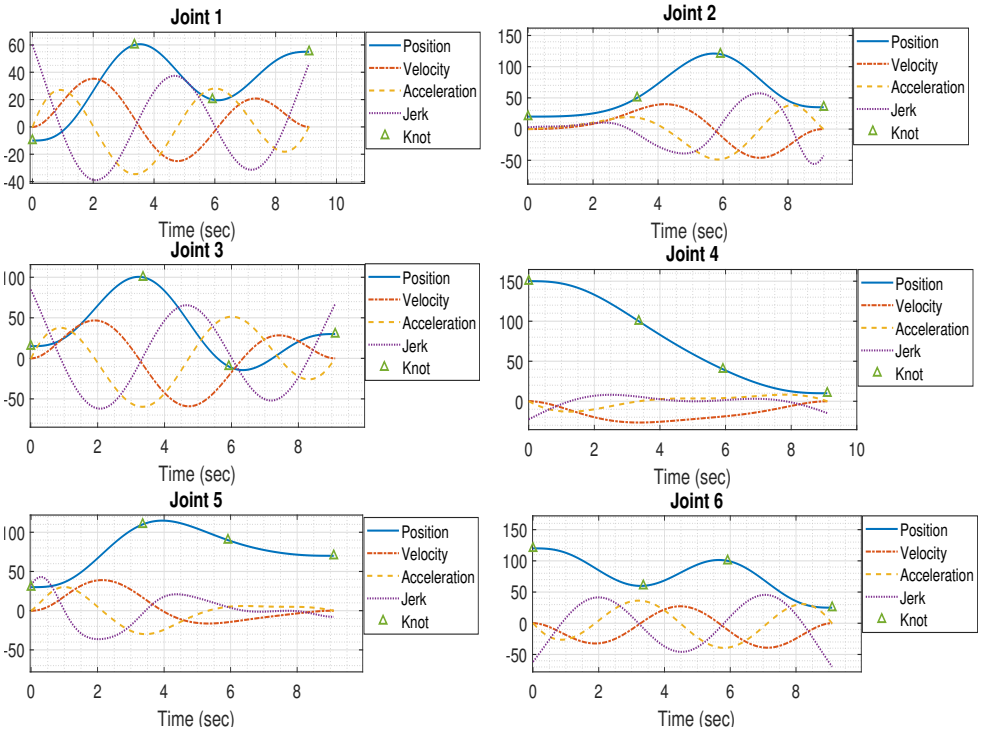

Fig. 5 Trajectory for minimum time-jerk CT

5 Optimal time-jerk trajectory planning of a series-parallel robotic manipulator

With the validation from the previous test cases, this section presents the optimal time-jerk trajectory planning of an additive manufacturing task for the series-parallel robotic manipulator using GH-RBFs.

5.1 Structure and kinematics of the robotic manipulator

As discussed earlier, the structure of the RM in the current study is unique, the actuators are located at the base and use parallel linkages to actuate the upper arm of the RM. A schematic of the kinematic structure of the manipulator is shown in

Table 9 Mean values of the kinematic variables

Kinematic Variable	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Overall
Velocity (deg/sec) (sec)	16.1554	20.5497	26.9163	15.3817	14.2446	19.5593	18.8012
Acceleration (deg/sec^2)	17.8169	18.8787	29.5047	5.8334	12.1645	21.6925	17.6484
Jerk (deg/sec^3)	23.7875	23.4040	38.5119	4.6917	14.5361	29.2835	22.3691

Figure 6. The forward and inverse kinematics of RM can be derived by solving the closed-loop equations formed by the parallel linkages [32]. For this study, we considered the most widely used additive manufacturing infill, zigzag infill (shown in Figure 7), as the performance test case for TP with GH-RBFs.

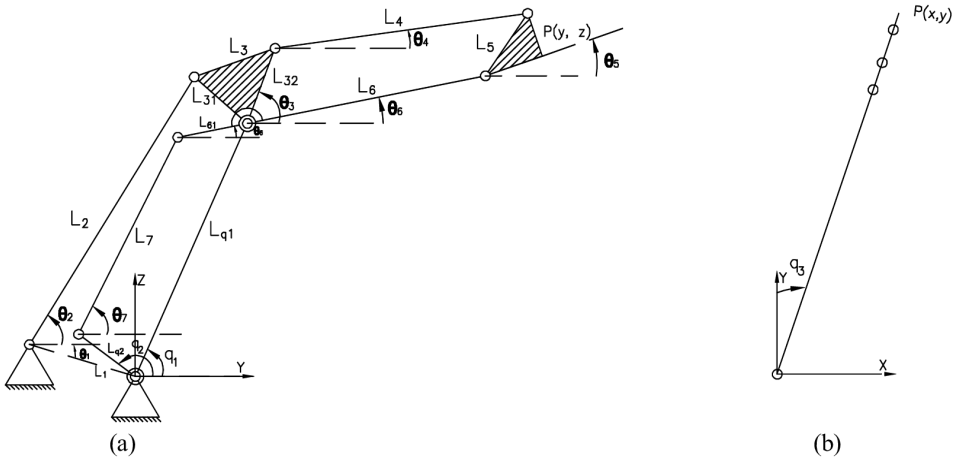

Fig. 6 a) Side view, and b) Top view of the kinematic structure of the robotic manipulator

5.2 Optimal trajectory planning

Unlike the previous test cases, we attempt to solve the TP as a multi-objective design optimization problem using the NSGA-II algorithm in MATLAB. The outcome is a set of non-dominating solutions, also known as *pareto front*, rather than a single solution. Moreover, one chooses a solution based on the trade-off between the two objectives given the application. In the present study, the test case from the ES is transformed into JS for the TP, the corresponding actuator angles and constraints are listed in Table 10 and 11. To compare the results with the previous studies, null boundary conditions are assigned only for velocity while acceleration and jerk are free. Figure 8 shows the optimal Pareto fronts for GH-RBFs and CS, and the joint trajectories for the minimum time solution using GH-RBFs.

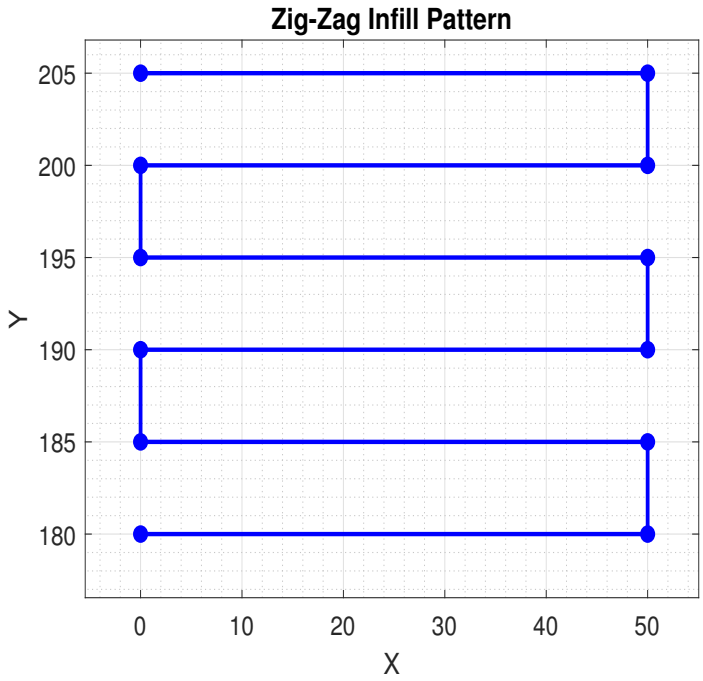

Fig. 7 Zig-zag infill pattern

Table 10 Knots for Minimum time-jerk Trajectory Planning in JS

Knot	Joint (deg.)		
	q_1	q_2	q_3
1	84.67	161.94	0
2	81.63	161.59	15.52
3	79.44	161.24	15.12
4	82.45	161.70	0
5	80.19	161.36	0
6	77.18	160.79	14.74
7	74.86	160.25	14.38
8	77.87	160.94	0
9	75.49	160.40	0
10	72.46	159.58	14.04
11	69.96	158.79	13.71
12	73.039	159.793	0

The velocity, acceleration, and jerk profiles are continuous satisfying the kinematic constraints and boundary conditions. For the same minimal total time ($t_f = 32.3784\ s$), the jerk index obtained using GH-RBFs is nearly 51% lesser than that of CS. Furthermore, for the same minimal jerk index ($JI = 2.2\ deg/sec^2$), the total time obtained using GH-RBFs is nearly 15% lesser than that of CS. This illustrates the

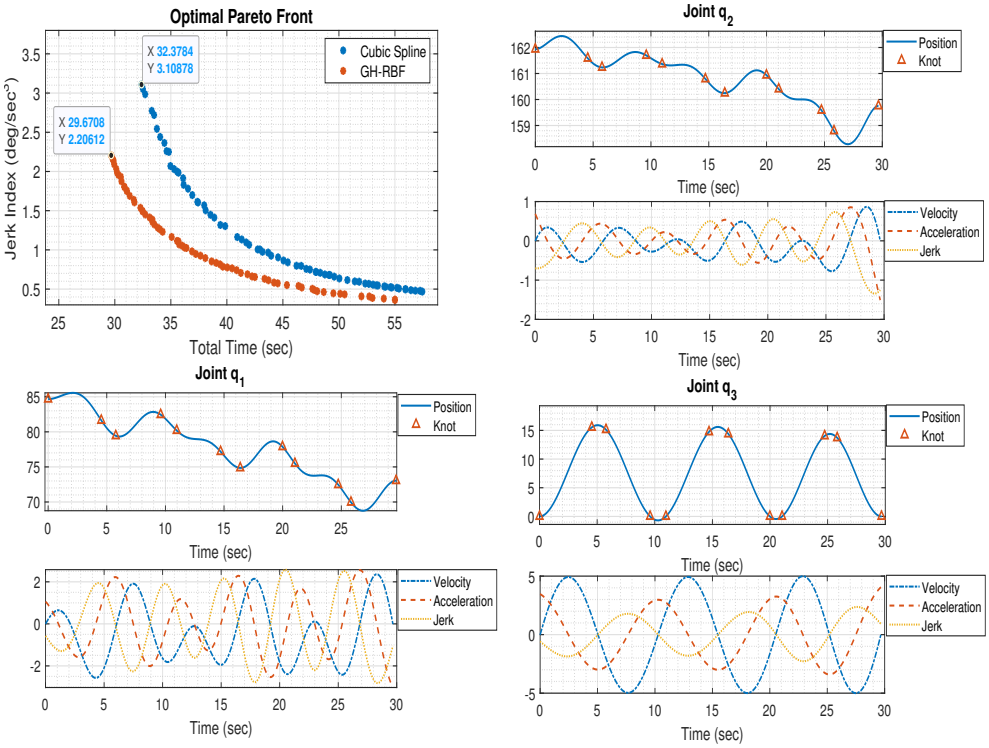

Fig. 8 Optimal pareto front and trajectory for minimum time solution using GH-RBFs

Table 11 Knots for Minimum time-jerk Trajectory Planning in JS

Constraints	Joint		
	q_1	q_2	q_3
Velocity (deg/sec)	5	5	5
Acceleration (deg/sec^2)	5	5	5
Jerk (deg/sec^3)	5	5	5

superior performance of GH-RBFs over the other TP methods and can be used for planning minimal-time smooth trajectories for the tasks of series-parallel RMs.

6 Conclusions

The current study introduces a new TP method to generate the optimal time-jerk trajectories for a series-parallel hybrid robotic manipulator. The new approach is validated against the various types of test cases before implementing on the series-parallel hybrid RM. The TP is performed in the joint space of the RM by converting it into a non-linear programming (NLP) problem. Furthermore, the objective function of the

NLP is the algebraic sum of the two conflicting parameters, total time and jerk index. However, for the final test case using series-parallel hybrid RM, the TP problem is solved as a multi-objective optimization problem using NSGA-II.

The major findings of the current study can be listed as follows,

1. A new RBF, known as Generalized Hyperbolic Radial Basis Function (GH-RBF) is introduced based on the minimal jerk requirement of the TP.
2. Furthermore, the GH-RBFs are conditionally positive definite for non-negative integer exponents for $x \in \mathbb{R}^1$, similar to generalized multi-quadric radial basis functions (GMQ-RBFs).
3. The GH-RBFs can generate trajectories that are comparable and better than those obtained using standard techniques like CS, B-splines, Gaussian and MQ-RBFs, NN, etc.
4. The GH-RBFs yielded minimum jerk trajectories for all the test cases (both CTs and PPTs) including the case with series-parallel hybrid RM, compared to other methods.
5. The GH-RBFs can be employed to generate optimal time-jerk trajectories for series-parallel hybrid RM.

The GH-RBFs can be employed in the approximation problems, like GMQ-RBFs, moreover, for a negative exponent, the GH-RBFs performed better than the GMQ-RBFs. The definition of GH-RBFs can be extended to $x \in \mathbb{R}^n$ which have huge scope in approximation theory, machine learning, control theory, etc. The future work focus on improving the optimization algorithms to decrease the computational time for the TP. In addition, the scope of the GH-RBFs can be extended for planning smooth trajectories of autonomous vehicles, spacecrafts, re-entry vehicles, etc.

Declarations

Author contribution. Deva Karthik Lakshman Dasu: worked on the basic idea, mathematical proofs, code, and drafted the manuscript. Anirudh R: contributed in critical review of the methodologies used in the study. Manu V Unnithan: reviwed and approved the final manuscript.

Funding. The current study is supported by Vikram Sarabhai Space Center (VSSC).

Conflict of interest. The authors declare no competing interests.

References

[1] Liu, X.-J., Li, J., Zhou, Y.: Kinematic optimal design of a 2-degree-of-freedom 3-parallelogram planar parallel manipulator. Mechanism and Machine Theory **87**, 1–17 (2015) https://doi.org/10.1016/j.mechmachtheory.2014.12.014

[2] Kong, M.X., You, W., Du, Z.J., Sun, L.N.: Optimal design for a 2-dof high dynamic manipulator based on parallelogram mechanism. In: 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 242–247 (2010). https://doi.org/10.1109/AIM.2010.5695735

[3] Cong, V.D., Phuong, L.: Design and fabrication of 3-dof robot arm usingparallelogram mechanisms. International Journal of Emerging Trends & Technology in Computer Science **9**, 1224–1229 (2021) https://doi.org/10.30534/ijeter/2021/03992021

[4] Sanjuan, J., Rahman, M., Rulik, I.: Forward kinematic analysis of dobot using closed-loop method. IAES International Journal of Robotics and Automation (IJRA) **9**, 153–159 (2020) https://doi.org/10.11591/ijra.v9i3.pp153-159

[5] Papadopoulos, E., Abu-Abed, A.: On the Design of Zero Reaction Manipulators. Journal of Mechanical Design **118**(3), 372–376 (1996) https://doi.org/10.1115/1.2826895

[6] Almasri, E., Uyguroğlu, M.K.: Trajectory optimization in robotic applications, survey of recent developments. Preprints (2021) https://doi.org/10.20944/preprints202105.0281.v1

[7] Kahn, M.E., Roth, B.: The Near-Minimum-Time Control Of Open-Loop Articulated Kinematic Chains. Journal of Dynamic Systems, Measurement, and Control **93**(3), 164–172 (1971) https://doi.org/10.1115/1.3426492

[8] Bobrow, J.E., Dubowsky, S., Gibson, J.S.: Time-optimal control of robotic manipulators along specified paths. The International Journal of Robotics Research **4**(3), 3–17 (1985) https://doi.org/10.1177/027836498500400301

[9] Kim, B.K., Shin, K.G.: Minimum-time path planning for robot arms and their dynamics. IEEE Transactions on Systems, Man, and Cybernetics **SMC-15**(2), 213–223 (1985) https://doi.org/10.1109/TSMC.1985.6313351

[10] Lin, C., Chang, P., Luh, J.: Formulation and optimization of cubic polynomial joint trajectories for industrial robots. IEEE Transactions on Automatic Control **28**(12), 1066–1074 (1983) https://doi.org/10.1109/TAC.1983.1103181

[11] Wang, C.-H., Horng, J.-G.: Constrained minimum-time path planning for robot manipulators via virtual knots of the cubic b-spline functions. IEEE Transactions on Automatic Control **35**(5), 573–577 (1990) https://doi.org/10.1109/9.53526

[12] Shin, K., Mckay, N.: Minimum-time trajectory planning for industrial robots with general torque constraints. In: Proceedings. 1986 IEEE International Conference on Robotics and Automation, vol. 3, pp. 412–417 (1986). https://doi.org/10.1109/ROBOT.1986.1087662

[13] Tse, K.-M., Wang, C.-H.: Evolutionary optimization of cubic polynomial joint trajectories for industrial robots. In: SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218), vol. 4, pp. 3272–32764 (1998). https://doi.org/10.1109/ICSMC.1998.726508

[14] Abu-Dakka, F., Assad, I., Alkhdour, R., Abderrahim, M.: Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots. The International Journal of Advanced Manufacturing Technology **89**, 389–406 (2017) https://doi.org/10.1007/s00170-016-9050-1

[15] Piazzi, A., Visioli, A.: Global minimum-jerk trajectory planning of robot manipulators. IEEE Transactions on Industrial Electronics **47**(1), 140–149 (2000) https://doi.org/10.1109/41.824136

[16] Piazzi, A., Visioli, A.: An interval algorithm for minimum-jerk trajectory planning of robot manipulators. In: Proceedings of the 36th IEEE Conference on Decision and Control, vol. 2, pp. 1924–19272 (1997). https://doi.org/10.1109/CDC.1997.657874

[17] Kyriakopoulos, K.J., Saridis, G.N.: Minimum jerk path generation. In: Proceedings. 1988 IEEE International Conference on Robotics and Automation, pp. 364–3691 (1988). https://doi.org/10.1109/ROBOT.1988.12075

[18] Simon, D.: The application of neural networks to optimal robot trajectory planning. Robotics and Autonomous Systems **11**(1), 23–34 (1993) https://doi.org/10.1016/0921-8890(93)90005-W

[19] Lin, H.-I., Liu, Y.-C.: Minimum-jerk robot joint trajectory using particle swarm optimization. In: 2011 First International Conference on Robot, Vision and Signal Processing, pp. 118–121 (2011). https://doi.org/10.1109/RVSP.2011.70

[20] Lin, H.-I.: A fast and unified method to find a minimum-jerk robot joint trajectory using particle swarm optimization. Journal of Intelligent & Robotic Systems **75**(3), 379–392 (2014) https://doi.org/10.1007/s10846-013-9982-8

[21] Cao, B., Dodds, G.I.: Time-optimal and smooth joint path generation for robot manipulators. In: 1994 International Conference on Control - Control '94., vol. 2, pp. 1122–11272 (1994). https://doi.org/10.1049/cp:19940293

[22] Gasparetto, A., Zanotto, V.: A new method for smooth trajectory planning of robot manipulators. Mechanism and Machine Theory **42**(4), 455–471 (2007) https://doi.org/10.1016/j.mechmachtheory.2006.04.002

[23] Gasparetto, A., Zanotto, V.: A technique for time-jerk optimal planning of robot trajectories. Robotics and Computer-Integrated Manufacturing **24**(3), 415–426 (2008) https://doi.org/10.1016/j.rcim.2007.04.001

[24] Gasparetto, A., Zanotto, V.: Optimal trajectory planning for industrial robots. Advances in Engineering Software **41**(4), 548–556 (2010) https://doi.org/10.1016/j.advengsoft.2009.11.001

[25] Gasparetto, A., Lanzutti, A., Vidoni, R., Zanotto, V.: Experimental validation

and comparative analysis of optimal time-jerk algorithms for trajectory planning. Robotics and Computer-Integrated Manufacturing **28**(2), 164–181 (2012) https://doi.org/10.1016/j.rcim.2011.08.003

[26] Lu, S., Zhao, J., Jiang, L., Liu, H.: Solving the time-jerk optimal trajectory planning problem of a robot using augmented lagrange constrained particle swarm optimization. Mathematical Problems in Engineering **2017**(1), 1921479 (2017) https://doi.org/10.1155/2017/1921479

[27] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002) https://doi.org/10.1109/4235.996017

[28] Chettibi, T.: Smooth point-to-point trajectory planning for robot manipulators by using radial basis functions. Robotica **37**(3), 539–559 (2019) https://doi.org/10.1017/S0263574718001169

[29] Huang, Y., Fei, M.: Motion planning of robot manipulator based on improved nsga-ii. International Journal of Control, Automation and Systems **16**(4), 1878–1886 (2018) https://doi.org/10.1007/s12555-016-0693-3

[30] Huang, J., Hu, P., Wu, K., Zeng, M.: Optimal time-jerk trajectory planning for industrial robots. Mechanism and Machine Theory **121**, 530–544 (2018) https://doi.org/10.1016/j.mechmachtheory.2017.11.006

[31] 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. Mechanism and Machine Theory **137**, 127–153 (2019) https://doi.org/10.1016/j.mechmachtheory.2019.03.019

[32] Lakshman, D.D.K., R, A., Unnithan, M.: Optimal smooth trajectory planning of robotic manipulator for additive manufacturing in space. In: 13th International Conference & Exhibition on "Innovations & Development in High Altitude Technologies & Related Mechanisms", Agra, India, pp. 219–225 (2023)

[33] Wang, Z., Li, Y., Shuai, K., Zhu, W., Chen, B., Chen, K.: Multi-objective trajectory planning method based on the improved elitist non-dominated sorting genetic algorithm. Chinese Journal of Mechanical Engineering **35**(1), 7 (2022) https://doi.org/10.1186/s10033-021-00669-x

[34] Liu, Y., Liang, L., Han, H., Zhang, S.: A method of energy-optimal trajectory planning for palletizing robot. Mathematical Problems in Engineering **2017**(1), 5862457 (2017) https://doi.org/10.1155/2017/5862457

[35] Liu, X., Qiu, C., Zeng, Q., Li, A., Xie, N.: Time-energy optimal trajectory planning for collaborative welding robot with multiple manipulators. Procedia Manufacturing **43**, 527–534 (2020) https://doi.org/10.1016/j.promfg.2020.02.

174 . Sustainable Manufacturing - Hand in Hand to Sustainability on Globe: Proceedings of the 17th Global Conference on Sustainable Manufacturing

[36] Wu, G., Zhao, W., Zhang, X.: Optimum time-energy-jerk trajectory planning for serial robotic manipulators by reparameterized quintic nurbs curves. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science **235**(19), 4382–4393 (2021) https://doi.org/10.1177/0954406220969734

[37] Kucuk, S.: Optimal trajectory generation algorithm for serial and parallel manipulators. Robotics and Computer-Integrated Manufacturing **48**, 219–232 (2017) https://doi.org/10.1016/j.rcim.2017.04.006

[38] Visioli, A.: Trajectory planning of robot manipulators by using algebraic and trigonometric splines. Robotica **18**(6), 611–631 (2000) https://doi.org/10.1017/S0263574700002721

[39] Scheiderer, C., Thun, T., Meisen, T.: Bézier curve based continuous and smooth motion planning for self-learning industrial robots. Procedia Manufacturing **38**, 423–430 (2019) https://doi.org/10.1016/j.promfg.2020.01.054 . 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2019), June 24-28, 2019, Limerick, Ireland, Beyond Industry 4.0: Industrial Advances, Engineering Education and Intelligent Manufacturing

[40] 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. The International Journal of Advanced Manufacturing Technology **120**(1), 297–312 (2022) https://doi.org/10.1007/s00170-022-08696-1

[41] Mirinejad, H., Inanc, T.: An rbf collocation method for solving optimal control problems. Robotics and Autonomous Systems **87**, 219–225 (2017) https://doi.org/10.1016/j.robot.2016.10.015

[42] Mirinejad, H., Inanc, T., Zurada, J.M.: Radial basis function interpolation and galerkin projection for direct trajectory optimization and costate estimation. IEEE/CAA Journal of Automatica Sinica **8**(8), 1380–1388 (2021) https://doi.org/10.1109/JAS.2021.1004081

[43] SAVITHA, R., SURESH, S., SUNDARARAJAN, N.: A fully complex-valued radial basis function network and its learning algorithm. International Journal of Neural Systems **19**(04), 253–267 (2009) https://doi.org/10.1142/S0129065709002026 . PMID: 19731399

[44] Savitha, R., Suresh, S., Sundararajan, N., Kim, H.J.: A fully complex-valued radial basis function classifier for real-valued classification problems. Neurocomputing **78**(1), 104–110 (2012) https://doi.org/10.1016/j.neucom.2011.05.036 . Selected papers from the 8th International Symposium on Neural Networks

(ISNN 2011)

[45] Mitaim, S., Kosko, B.: The shape of fuzzy sets in adaptive function approximation. IEEE Transactions on Fuzzy Systems **9**(4), 637–656 (2001) https://doi.org/10.1109/91.940974

[46] Wendland, H.: Scattered Data Approximation. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (2004)

[47] Fasshauer, G.F.: Meshfree Approximation Methods with MATLAB. World Scientific Publishing Co., Inc., USA (2007)

[48] Saleh, R.A., Saleh, A.K.M.E.: Statistical Properties of the log-cosh Loss Function Used in Machine Learning (2024). https://arxiv.org/abs/2208.04564

[49] Balogh, C.B.: Asymptotic expansions of the modified bessel function of the third kind of imaginary order. SIAM Journal on Applied Mathematics **15**(5), 1315–1323 (1967) https://doi.org/10.1137/0115114

[50] Olver, F.W.J., Daalhuis, A.B.O., Lozier, D.W., Schneider, B.I., Boisvert, R.F., Clark, C.W., B. R. Mille and, B.V.S., Cohl, H.S., M. A. McClain, e.: NIST Digital Library of Mathematical Functions. http://dlmf.nist.gov/, Release 1.2.2 of 2024-09-15 (2020). http://dlmf.nist.gov/