Robotica (2025), **43**, pp. 4384–4401
doi:10.1017/S0263574725102890

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

An adaptive path planning algorithm with singularity consistency for robotic arms

Xianyou Zhong, Mengjiao Shen, Xiao Lin, Chengju Liu and Qijun Chen

College of Electronic and Information Engineering, Tongji University, Shanghai, China
Corresponding author: Qijun Chen; Email: qjchen@tongji.edu.cn

Received: 27 April 2025; **Revised:** 1 October 2025; **Accepted:** 5 November 2025; **First published online:** 3 December 2025

Keywords: singularity consistency; path arc-length parameterization; weights adaptive sampling

Abstract

The study presents a novel approach to address challenges posed by singularities in robotic arm motion, focusing on Cartesian path planning and geometric path adherence. Recognizing limitations in traditional singularity avoidance methods, the research proposes a comprehensive strategy: reconstructing motion patterns in singular regions through singularity-consistent representations, applying arc-length reparameterization to Cartesian geometric paths, and incorporating path curvature as a dynamic weighting factor for sampling interval adjustment. This method achieves a balance between joint velocity smoothness and geometric tracking accuracy in Cartesian space, significantly enhancing the robot's ability to adhere to prescribed geometric paths, particularly near singularities. Experimental results demonstrate the efficacy of the proposed approach in facilitating smooth singularity transitions, improving joint velocity continuity, and enhancing geometric path adherence. The study contributes to robotic arm path planning by offering a practical solution for applications requiring precise trajectory following and effective singularity handling.

1. Introduction

Robotic arms have become essential in various fields due to their precision and flexibility. However, a persistent challenge in robotic motion planning lies in addressing singularities [1, 2] – configurations where the Jacobian matrix loses rank, leading to potential velocity fluctuations and control instabilities. Foundational singularity-theoretic and algebraic treatments further clarify these configurations and their geometry [3, 4]. Maintaining geometric path adherence, that is, the robot's ability to precisely follow prescribed Cartesian paths, serves as a crucial constraint for high-precision robotic tasks. Near singular configurations, even small joint variations can cause significant end-effector deviations, jeopardizing path accuracy; quantitative "closeness-to-singularity" measures provide early warning signals for such risks [5, 6]. As robotic applications expand in advanced manufacturing and medical robotics, developing strategies that effectively navigate singularities while ensuring high-fidelity path tracking remains a critical research imperative.

Over decades, singularity avoidance methods have evolved mainly along two lines: *Jacobian matrix optimization* and *joint velocity adjustment*. The first line includes techniques such as the pseudoinverse [7, 8], damped least squares method [9, 10], adaptive damping [11, 12], and singular value decomposition (SVD)-based approaches [13–15] that maintain full rank Jacobians near singularities; algebraic analyses also support reliable singularity detection across architectures [4]. Although successful in preserving mathematical stability, these methods often introduce deviations from the intended Cartesian path, reducing task accuracy. The second line focuses on modulating joint velocities to traverse singular regions, including singularity-consistent methods [16, 17], null-space projection techniques [18, 19], and velocity scaling [20]. Recent work on trajectory optimization [21, 22] attempts to minimize path deviations and improve efficiency. In parallel, studies relating workspace structure,

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

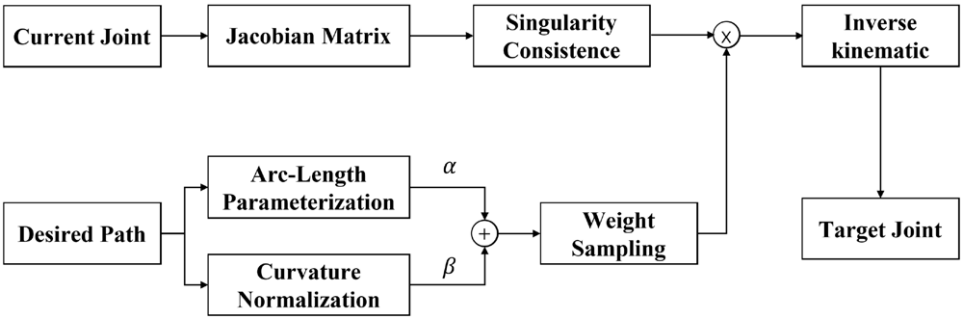


Figure 1. *Block diagram of framework.*

redundancy, and singularity loci highlight design–control trade-offs that can be exploited for avoidance [31]. Despite these advances, the increasing demand for precise geometric path adherence reveals limitations of existing singularity avoidance methods, especially when path deviations or efficiency losses are unacceptable.

Complementing singularity handling, techniques for geometric path adherence focus on accurate path representation and control. Path parameterization methods optimize trajectory descriptions to facilitate tracking, including predictive approaches [23], dynamic parameterization [24], and time-optimal methods [25, 26], all enhancing robot responsiveness and constraint compliance. On the control side, diverse tracking strategies address path deviation minimization: singularity-consistent tracking [16, 27], coordinate transformation-based contour following [28], adaptive fuzzy controllers [29, 30], neural network inverse kinematics [32, 33], and fractional order fuzzy PID control [34, 35]. In addition, metric-based proximity assessment [5, 6] and algebraic singularity tools [4] offer complementary constraints for path planners, while design-time shaping of singularity-free workspaces provides a structural route to avoidance [36].

To address these intertwined challenges, we propose a novel framework integrating singularity modeling, arc-length parameterization, and curvature-based dynamic sampling. As illustrated in Figure 1, our approach consists of two complementary components: (1) a singularity-consistent method that evaluates motion difficulty and adaptively adjusts joint velocities for smooth transitions near singularities and (2) a curvature-adaption method that calculates trajectory curvature, determines adaptive sampling density, and performs arc-length parameterization for refined path resampling. By jointly handling singularities and complex curved trajectories, this dual approach overcomes the limitations of traditional single-focus methods, achieving enhanced motion smoothness and trajectory accuracy across diverse robotic tasks.

The key contributions of our approach are as follows:

- We integrate singularity-consistent theory with arc-length parameterization, enabling accurate modeling of Cartesian paths in joint space and ensuring seamless transitions through singular configurations.
- We propose a curvature-based adaptive sampling method that utilizes path curvature as a weighting factor to adaptively adjust sampling intervals. This approach effectively balances joint velocity smoothness with Cartesian tracking accuracy, particularly in regions of high curvature or near singularities.

This paper is organized as follows: Section 2 presents the proposed methodology, including the singularity-consistency method and the curvature-adaption method (with details on path parameterization, curvature modeling, and adaptive sampling). Section 3 describes the experimental setup and the conducted experiments. Finally, Section 4 concludes the paper, summarizing the findings and outlining future research directions.

2. Methodology

2.1. Singularity-consistent method

For a serial-link manipulator, its motion state is defined by n joint variables $\mathbf{q} = (q_1, q_2, \ldots, q_n)$, which determine the configuration of the manipulator. On the other hand, the pose of the end-effector,

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

including its position and orientation, can be described by m independent variables $\mathbf{r} = (r_1, r_2, \ldots, r_m)$. In most practical cases, $m = 6$, as the end-effector pose typically consists of three-dimensional position and three-dimensional orientation, mathematically represented as $\mathrm{SE}(3) = \mathbb{R}^3 \times \mathrm{SO}(3)$, the space of Euclidean motions. In certain scenarios, such as when only the position is considered or the motion is constrained to a plane, the dimensionality of $\mathbf{r}$ can be further reduced, for example, $\mathbf{r} \in \mathbb{R}^3$ or $\mathbb{R}^2$.

Let $\mathbf{q} \in \mathbb{R}^n$ be joint variables and $\mathbf{r} \in \mathbb{R}^m$ be task-space variables (position/orientation under a chosen parameterization). The forward kinematics is

$$\mathbf{r} = \mathrm{fk}(\mathbf{q}) \tag{1}$$

where $\mathbf{r}$ represents the pose of the end-effector (e.g., position and orientation) and $\mathbf{q}$ represents the joint variables of the manipulator.

When the end-effector follows a predefined trajectory, its pose $\mathbf{r}$ can also be expressed as a function of **the path parameter** p, which typically represents time or a normalized progress variable along the trajectory:

$$\mathbf{r} = \mathbf{r}(p). \tag{2}$$

Jacobian matrices are key to describing the mapping between joint velocities and Cartesian end-effector velocities. This relationship can be expressed as:

$$\dot{\mathbf{r}} = J(\mathbf{q})\dot{\mathbf{q}}, \tag{3}$$

where $J(\mathbf{q})$ is the manipulator Jacobian, defined as:

$$J(\mathbf{q}) = \frac{\partial \mathrm{fk}(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{m \times n}. \tag{4}$$

The Jacobian $J(\mathbf{q})$ provides a linear mapping between the joint space velocity $\dot{\mathbf{q}}$ and the Cartesian space velocity $\dot{\mathbf{r}}$. This relationship is fundamental in robotic motion control, enabling the computation of the end-effector velocity from joint velocities, and vice versa.

We mark the unit vector tangent of $\mathbf{r}(p)$ as:

$$\mathbf{S}(p) = \frac{\partial \mathbf{r}(p)}{\partial p} \in \mathbb{R}^m, \ \|\mathbf{S}(p)\| = 1. \tag{5}$$

To describe this relationship, we define the following augmented kinematic function:

$$\mathbf{h}(\bar{\mathbf{q}}) = \mathrm{fk}(\mathbf{q}) - \mathbf{r}(p), \tag{6}$$

where $\bar{\mathbf{q}} = [\mathbf{q}^\top, p]^\top$ is an extended parameter vector that includes both the joint variables $\mathbf{q}$ and the path parameter p. In the ideal case, $\mathbf{h}(\bar{\mathbf{q}}) \equiv \mathbf{0}$, which implies that the end-effector trajectory follows the desired path precisely.

Linearizing at $(\mathbf{q}, p)$ yields

$$\mathbf{H}(\bar{\mathbf{q}})\, d\bar{\mathbf{q}} = \mathbf{0} \tag{7}$$

$$\mathbf{H}(\bar{\mathbf{q}}) = \frac{\partial \mathbf{h}}{\partial \bar{\mathbf{q}}} = \begin{bmatrix} J(\mathbf{q}) & -\mathbf{S}(p) \end{bmatrix} \in \mathbb{R}^{m \times (n+1)} \tag{8}$$

Here, $\mathbf{H}(\bar{\mathbf{q}})$ denotes the augmented matrix of the original Jacobian. It integrates the path parameter p into the standard Jacobian framework, providing a unified representation for analyzing and controlling the end-effector's trajectory tracking.

For computational convenience, we express the general solution of Eq. (7) in the following form:

$$d\bar{\mathbf{q}} = b\, \bar{\mathbf{F}}(\bar{\mathbf{q}}), \tag{9}$$

where b is an arbitrarily small scalar and $\bar{\mathbf{F}}(\bar{\mathbf{q}}) \in \ker \mathbf{H}(\bar{\mathbf{q}}) \subset \mathbb{R}^{n+1}$ is a vector in the null space of the augmented Jacobian matrix $\mathbf{H}(\bar{\mathbf{q}})$.

According to matrix theory, the solution structure depends on the relationship between m and n.

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

Special case: $m = n$

Here, $J \in \mathbb{R}^{n\times n}$ is square. Consider

$$\bar{\mathbf{F}}_{\square}(\bar{\mathbf{q}}) = \begin{bmatrix} \operatorname{adj} J(\mathbf{q})\, \mathbf{S}(p) \\ \det J(\mathbf{q}) \end{bmatrix} \in \mathbb{R}^{n+1}. \tag{10}$$

where the subscript $\square$ emphasizes that this form applies specifically to the square-Jacobian case ($m = n$). Using $J\ \operatorname{adj} J = \det(J)\, \mathbf{I}$,

$$\mathbf{H}\, \bar{\mathbf{F}}_{\square} = J\ \operatorname{adj} J\, \mathbf{S} - \det(J)\, \mathbf{S} = \big(\det(J)\, \mathbf{I} - \det(J)\, \mathbf{I}\big)\mathbf{S} = \mathbf{0},$$

thus $\bar{\mathbf{F}}_{\square} \in \ker \mathbf{H}$. Define the singularity-consistent first-order step

$$d\bar{\mathbf{q}} = b\, \bar{\mathbf{F}}_{\square}(\bar{\mathbf{q}}), \qquad b \in \mathbb{R} \text{ small}, \tag{11}$$

which automatically satisfies (7). As $\det J \to 0$, (11) yields $dp = b\ \det J \to 0$, avoiding blow-ups in $d\mathbf{q}$ or dp – the desired "singularity-consistent" behavior.

Properties. If $\operatorname{rank} J = n - 1$, then $\operatorname{adj} J$ has rank 1 and $\det J = 0$; (11) reduces to $dp = 0$ with a nonzero $d\mathbf{q} \in \ker J$. If $\operatorname{rank} J \leq n - 2$, then $\operatorname{adj} J = \mathbf{0}$ and a numerical null space (SVD) of $\mathbf{H}$ is required.

General case: $m \neq n$

To avoid the square-Jacobian assumption behind $\det(J)$, define

$$\mathbf{A}(\mathbf{q}) = J(\mathbf{q})\, J^{\top}(\mathbf{q}) \in \mathbb{R}^{m\times m}, \qquad \alpha(\mathbf{q}) = \det\big(\mathbf{A}(\mathbf{q})\big)\ (\geq 0), \tag{12}$$

and construct

$$\bar{\mathbf{F}}(\bar{\mathbf{q}}) = \begin{bmatrix} J^{\top}(\mathbf{q})\ \operatorname{adj}\big(\mathbf{A}(\mathbf{q})\big)\, \mathbf{S}(p) \\ \det\big(\mathbf{A}(\mathbf{q})\big) \end{bmatrix} \in \mathbb{R}^{n+1}. \tag{13}$$

Since $\mathbf{A}\ \operatorname{adj} \mathbf{A} = \det(\mathbf{A})\, \mathbf{I}_m$,

$$\mathbf{H}\, \bar{\mathbf{F}} = J\, J^{\top} \operatorname{adj}(\mathbf{A})\, \mathbf{S} - \det(\mathbf{A})\, \mathbf{S} = \big(\mathbf{A} \operatorname{adj} \mathbf{A} - \det(\mathbf{A})\mathbf{I}_m\big)\mathbf{S} = \mathbf{0},$$

hence $\bar{\mathbf{F}} \in \ker \mathbf{H}$ and

$$d\bar{\mathbf{q}} = b\, \bar{\mathbf{F}}(\bar{\mathbf{q}}), \qquad b \in \mathbb{R} \text{ small}. \tag{14}$$

Consistency with the square case. When $m = n$, using $\operatorname{adj}(JJ^{\top}) = \operatorname{adj}(J)^{\top} \operatorname{adj}(J)$ together with $J^{\top} \operatorname{adj}(JJ^{\top}) = \det(J)\ \operatorname{adj}(J)$ and $\det(JJ^{\top}) = \det(J)^2$, we obtain

$$\bar{\mathbf{F}} = \det(J) \begin{bmatrix} \operatorname{adj} J\, \mathbf{S} \\ \det J \end{bmatrix},$$

which is colinear with (10) and thus identical as a kernel direction.

Computation and robustness

(i) If $\mathbf{A}$ is nonsingular, use

$$\bar{\mathbf{F}} = \begin{bmatrix} \det(\mathbf{A})\, J^{\top} \mathbf{A}^{-1} \mathbf{S} \\ \det(\mathbf{A}) \end{bmatrix}$$

to avoid forming adjugates;

(ii) at generic singularities with $\operatorname{rank} J = m - 1$, $\operatorname{adj} \mathbf{A}$ has rank 1 and $\det \mathbf{A} = 0$, yielding $dp = 0$ and a nonzero feasible $d\mathbf{q}$;

(iii) at higher-order singularities ($\operatorname{rank} J \leq m - 2$), $\operatorname{adj} \mathbf{A} = \mathbf{0}$, and one should compute a null vector of $\mathbf{H}$ via SVD;

(iv) with a joint space metric/weight $\mathbf{W} \succ 0$, take $\mathbf{A} = J\mathbf{W}J^{\top}$ and replace (13) accordingly.

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

Remark. The method only relies on the null-space relation $\mathbf{H}\,\bar{\mathbf{F}} = \mathbf{0}$ and never requires $\det(\mathbf{H})$. If a scalar conditioning index is desired, use $\mathbf{H}\mathbf{H}^{\top} = JJ^{\top} + \mathbf{S}\mathbf{S}^{\top}$ and inspect its singular values or $\det(\mathbf{H}\mathbf{H}^{\top})$.

By analyzing the structure of $\bar{\mathbf{F}}(\bar{\mathbf{q}})$, it becomes clear that $\mathbf{F}(\bar{\mathbf{q}})$ combines the joint angle variations and the path parameter p through the augmented parameter vector $\bar{\mathbf{q}}$. This coupling enables $\mathbf{F}(\bar{\mathbf{q}})$ playing a singularity-consistent role in the solution process. When the manipulator's joint configuration approaches a Jacobian singularity (i.e., $\det J(\mathbf{q}) \to 0$), the singularity is reflected in the augmented Jacobian matrix $\mathbf{H}(\bar{\mathbf{q}})$. Consequently, the system automatically adjusts the Cartesian sampling along the path parameter p, ensuring consistent behavior even in the presence of singularities.

By splitting Eq. (9) into two components, we obtain the following expressions for the joint differential motion and the path parameter differential:

$$d\mathbf{q} = b\,\mathbf{f}(\bar{\mathbf{q}}), \tag{15}$$

$$dp = b\ \det J(\mathbf{q})\ or\ b\ \det A(\mathbf{q}). \tag{16}$$

To ensure that the manipulator can traverse through the singular region while respecting the joint's maximum speed limits, we define the **joint velocity adjustment factor** b^* as the maximum allowable scaling factor for the joint velocities. Based on the current joint configuration and velocity limits, b^* is computed as:

$$b^* = \max\left\{b \mid b \cdot \|\mathbf{F}(\bar{\mathbf{q}})\| \le dq_{\max}\right\}, \tag{17}$$

where:

- $dq_{\max}$ represents the maximum allowable joint velocity, determined by the physical constraints of the manipulator.
- $\|\mathbf{F}(\bar{\mathbf{q}})\|$ is the norm of the joint velocity component derived from the mapping $\mathbf{F}(\bar{\mathbf{q}})$, which reflects the influence of both the Jacobian singularity and the path parameter p.

The adjustment factor b^* ensures that the joint velocities remain within their feasible limits, even as the manipulator approaches a singular configuration. When $\|\mathbf{F}(\bar{\mathbf{q}})\|$ becomes large near a singularity, b^* is automatically reduced to prevent the joint velocities from exceeding $dq_{\max}$. This mechanism allows the manipulator to pass through the singular region in a controlled and safe manner.

2.2. Curvature-adaption method

2.2.1. Path arc-length parameterization

While curves can be mathematically represented using path parameters, the variation of the cumulative arc length with respect to the path parameter is generally non-uniform. As illustrated in Figure 2, the arc length $L(t)$ of both Bézier and Lissajous curves is not linearly related to the parameter t (the red curve deviates significantly from the line $y = x$). The cumulative arc length $L(t)$ is defined as:

$$L(t) = \int_0^t ||\mathbf{C}'(u)||\, du \tag{18}$$

where $\mathbf{C}(t)$ represents the parametric equation of the curve. Due to the integral, uniform variation in t does not guarantee a uniform increase in $L(t)$. Consequently, uniform sampling of the path parameter t may result in irregular variations of the differential speed, approximated as $\frac{\Delta L}{\Delta t}$. As shown in the right subplots of Figure 2, the speed curves demonstrate significant and non-smooth fluctuations, indicating that uniform sampling in the parameter domain does not guarantee uniformity in the arc-length domain.

To address the non-uniformity challenge of the curve's arc length, a natural approach is to introduce the path arc length as the path parameter. This involves reparameterizing the path using the arc length s as the new parameter. For a path parameterized by σ in joint space, the arc length s is given by:

$$s = \int_0^{\sigma} \|\mathbf{q}'\|_M\, d\xi, \tag{19}$$

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

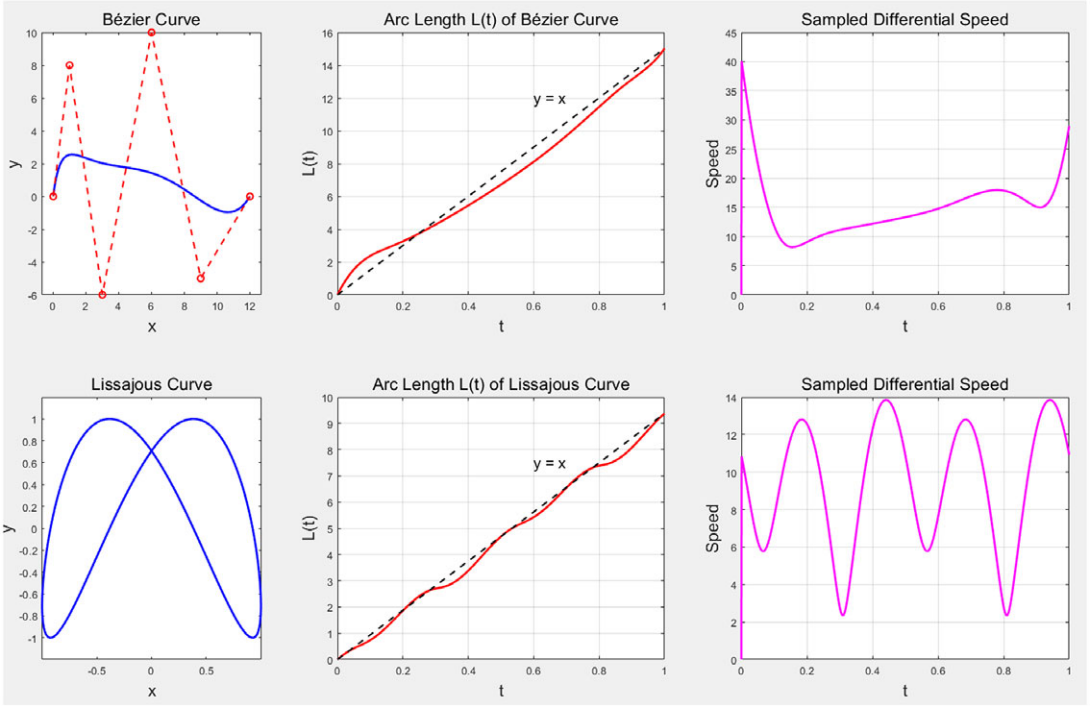


Figure 2. *Path arc length and speed profiles.* ***Top row:*** *Bézier curve example.* ***Bottom row:*** *Lissajous curve example.* ***Left:*** *The geometric paths (blue solid line) and their control points (red dashed line for the Bézier curve).* ***Middle:*** *The arc-length function $L(t)$ (red solid line) and the ideal linear relationship $y = x$ (black dashed line).* ***Right:*** *The corresponding sampled differential speed profile. The results demonstrate that the arc-length function $L(t)$ deviates from the ideal linear relationship, indicating a non-uniform parameterization with respect to arc length. This non-uniformity is also reflected in the fluctuating speed profiles.*

where $\|\mathbf{q}'\|_M = \sqrt{\mathbf{q}'^T M \mathbf{q}'}$ denotes the weighted Euclidean norm of the joint velocity, M is a metric matrix (assumed to be the identity matrix for simplicity), and $\mathbf{q}' = \frac{d\mathbf{q}}{d\sigma}$ represents the derivative of the joint coordinates with respect to the original path parameter.

By parameterizing the path with respect to arc length, we have:

$$\frac{ds}{d\sigma} = \|\mathbf{q}'\|_M. \tag{20}$$

This implies that equal increments in s correspond to equal distances traversed along the curve, regardless of the original parameterization. Consequently, sampling uniformly in s yields points that are equidistant along the path in joint space. This method, known as arc-length parameterization, effectively transforms a non-uniformly parameterized path into a uniformly parameterized one with respect to arc length, ensuring consistent and predictable motion along the curve. The resulting parameterization $\mathbf{q}(s)$ provides a more natural and intuitive representation of the path, facilitating tasks such as trajectory planning and control.

It is worth noting that for curves where analytical integration is difficult, a discrete sampling approach combined with numerical integration can be employed to approximate the arc length. Common methods for equidistant sampling include Simpson's rule, the Trapezoidal rule, and other numerical integration techniques, allowing us to obtain approximately uniformly spaced points with respect to arc length.

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

From this, we derive:

$$\frac{\partial s}{\partial \sigma} = \|\mathbf{q}'\|_M, \tag{21}$$

and consequently:

$$\frac{\partial \sigma}{\partial s} = \frac{1}{\|J^+\mathbf{h}\|_M} = \frac{1}{\|\mathbf{q}'\|_M}, \tag{22}$$

$$\frac{\partial \mathbf{q}}{\partial s} = \frac{J^+\mathbf{h}}{\|J^+\mathbf{h}\|_M} = \frac{\mathbf{q}'}{\|\mathbf{q}'\|_M}, \tag{23}$$

These equations describe how the arc length, the original parameter, and the joint space coordinates change with respect to each other. They are fundamental to understanding and implementing arc-length parameterization.

2.2.2. Curvature modeling

While arc-length parameterization provides a uniform representation of the geometric path in Cartesian space, another crucial geometric variable must be considered: the path curvature. Although sampling uniformly with respect to arc length ensures equidistant points along the path, neglecting the path's curvature can lead to subtle deformations in the robot's motion during trajectory tracking. This is because curvature directly influences the required joint accelerations and velocities. Consequently, ignoring curvature can compromise the robot's ability to precisely adhere to the intended geometric path, especially in regions with high curvature. Therefore, a comprehensive understanding of the path's behavior requires considering both arc length and curvature to ensure accurate and compliant motion.

For a given path parameterized by $\mathbf{q}(\sigma)$, the curvature κ in joint space can be defined using the second derivative of the path with respect to σ:

$$\kappa = \frac{\|\mathbf{q}' \times \mathbf{q}''\|_M}{\|\mathbf{q}'\|_M^3}, \tag{24}$$

where $\mathbf{q}'$ and $\mathbf{q}''$ are the first and second derivatives of $\mathbf{q}$ with respect to σ, respectively, and $\|\cdot\|_M$ denotes the norm induced by the metric matrix M. The numerator $\|\mathbf{q}' \times \mathbf{q}''\|_M$ represents the magnitude of the cross product of the tangent and acceleration vectors, capturing the "twist" of the curve, while the denominator $\|\mathbf{q}'\|_M^3$ normalizes this value relative to the cube of the tangent vector's magnitude.

The curvature κ provides a quantitative measure of how sharply the path bends at any point σ. High curvature values indicate regions of sharp turns or singularities, which may lead to large joint accelerations and potentially unstable behavior. Conversely, low curvature values correspond to smoother, more gradual transitions along the path.

To analyze the curvature effectively, it is important to examine its value range (curvature value domain) over the entire path. Let $\kappa_{\min}$ and $\kappa_{\max}$ denote the minimum and maximum curvature values along the path, respectively. The range $[\kappa_{\min}, \kappa_{\max}]$ provides critical information about the path's geometric complexity and the potential for dynamic instabilities. For practical implementations, it is often desirable to normalize the curvature to a unitless quantity for easier interpretation and comparison across different paths.

A curvature normalization function κ_{norm} can be defined as follows:

$$\kappa_{\text{norm}}(\sigma) = \frac{\kappa(\sigma) - \kappa_{\min}}{\kappa_{\max} - \kappa_{\min}}, \tag{25}$$

where $\kappa_{\text{norm}}(\sigma) \in [0, 1]$. This normalization maps the curvature values to a standardized range, enabling consistent evaluation and visualization of the path's geometric properties. Regions with $\kappa_{\text{norm}} \approx 0$ correspond to relatively straight segments, while $\kappa_{\text{norm}} \approx 1$ indicates areas with the sharpest bends.

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

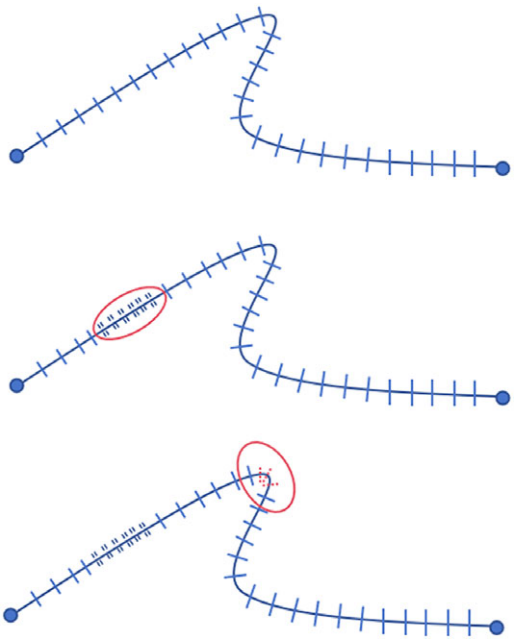


Figure 3. *Limitations of arc-length uniform sampling strategy. The figure illustrates potential issues arising from solely relying on arc length for sampling.* ***Top****: A planned path with uniform arc-length sampling.* ***Middle****: In regions with gentle curvature (circled area), the mapping between joint space and Cartesian space can lead to joint configurations near singularities, causing large joint movements for small end-effector displacements.* ***Bottom****: In areas of high curvature (circled area), uniform arc-length sampling may result in under-sampling, potentially missing crucial details of the path and compromising tracking accuracy.*

Furthermore, the normalized curvature can be incorporated into path planning and motion control algorithms to adaptively adjust speed or smooth transitions in regions of high curvature, thereby mitigating the risk of instability. By coupling curvature analysis with arc-length parameterization, a comprehensive understanding of the geometric and dynamic characteristics of the path can be achieved, leading to more robust and efficient motion planning strategies.

2.2.3. Weights adaptive sampling

As illustrated in Figure 3, path curvature significantly impacts trajectory tracking, particularly concerning the robot's ability to accurately follow the planned path. Neglecting curvature can lead to issues such as large joint movements near singularities in regions of low curvature, or under-sampling in areas of high curvature, ultimately compromising the robot's ability to adhere to the intended trajectory. To address these challenges and improve path fidelity, we introduce a curvature-adaptive weighted sampling algorithm. This algorithm is designed to ensure more accurate path adherence by dynamically adjusting the sampling density based on the curvature.

First, given a path represented as a sequence of discrete points $\mathbf{p}_i \in \mathbb{R}^2$, the curvature at each point is computed using three consecutive points. Let $\mathbf{p}_{\text{prev}}$, $\mathbf{p}_{\text{curr}}$, and $\mathbf{p}_{\text{next}}$ denote the previous, current, and next points, respectively. The vectors between these points are defined as:

$$\mathbf{v}_1 = \mathbf{p}_{\text{curr}} - \mathbf{p}_{\text{prev}}, \tag{26}$$

$$\mathbf{v}_2 = \mathbf{p}_{\text{next}} - \mathbf{p}_{\text{curr}}. \tag{27}$$

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

The curvature κ is then calculated as:

$$\kappa = \begin{cases} \frac{2 \cdot \|\mathbf{v}_1 \times \mathbf{v}_2\|}{\|\mathbf{v}_1\| \cdot \|\mathbf{v}_2\| \cdot \|\mathbf{v}_1 + \mathbf{v}_2\|}, & \text{if } \|\mathbf{v}_1\| \neq 0 \text{ and } \|\mathbf{v}_2\| \neq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

Here, $\|\cdot\|$ denotes the Euclidean norm, and $\times$ represents the cross product. If either $\mathbf{v}_1$ or $\mathbf{v}_2$ has zero magnitude, the curvature is set to zero.

To ensure that curvature values are within a standard range, they are normalized as follows:

$$\kappa_{\text{norm}} = \begin{cases} \frac{\kappa}{\max(\kappa)}, & \text{if } \max(\kappa) \neq 0, \\ 0, & \text{if } \max(\kappa) = 0. \end{cases} \tag{29}$$

Here, $\max(\kappa)$ represents the maximum curvature value across all points.

Then we design the weight function to integrate a base arc-length weight factor α with a curvature weight factor β, effectively combining these two crucial aspects.

$$w = \alpha + \beta \, \kappa_{\text{norm}}, \qquad \kappa_{\text{norm}} \in [0, 1]. \tag{30}$$

Here, α governs the baseline density along arc length (favoring more uniform spacing), whereas β increases sampling density in higher-curvature regions. Without loss of generality, we set $\alpha, \beta \geq 0$ and $\alpha + \beta = 1$ (the subsequent normalization makes only the relative balance matter). **In the subsequent experiments, we adopt $\boldsymbol{\alpha} = 0.6$ and $\boldsymbol{\beta} = 0.4$ as a balanced default.** In practical engineering applications, the parameters can be adjusted according to the specific task requirements and trajectory characteristics. For relatively flat trajectories where a smoother and more optimal velocity profile is desired, α can be tuned within [0.6, 0.8]; for tasks demanding high trajectory-following accuracy, such as precision machining or robotic path tracking in high-curvature segments, β can be tuned within [0.4, 0.6]. Nearby choices lead to similar qualitative behavior.

The cumulative weight $W(s)$ along the path is computed as the integral of the weight function with respect to the arc length s:

$$W(s) = \int_0^s w(s') \, ds'. \tag{31}$$

The cumulative weight is then normalized to the range [0, 1]:

$$W_{\text{norm}}(s) = \frac{W(s)}{\max(W(s))}. \tag{32}$$

To perform sampling based on the computed weights, we define a set of target weights uniformly distributed in [0, 1]:

$$W_{\text{target}} = \left\{ 0, \frac{1}{N-1}, \frac{2}{N-1}, \ldots, 1 \right\}, \tag{33}$$

where N is the number of desired sampling points. For each target weight, the closest cumulative weight is identified, and the corresponding path point is selected.

The following pseudocode summarizes the entire process:

To evaluate how well an executed path conforms to the planned trajectory, we employ the Fréchet distance as a metric for assessing their similarity. This metric provides the measure of adherence between the two paths. When the paths are represented as discrete sequences of points, the discrete Fréchet distance is commonly used. Let $P = \{p_1, p_2, \ldots, p_m\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$ denote two discrete curves in $\mathbb{R}^d$, representing the planned and executed paths, respectively. The discrete Fréchet distance is defined as:

$$d_F^d(P, Q) = \min_{\pi} \max_{(i,j) \in \pi} \|p_i - q_j\|, \tag{34}$$

where π represents all possible alignments (or couplings) of the two curves and $\|p_i - q_j\|$ is the Euclidean distance between points $p_i \in P$ and $q_j \in Q$. An alignment π is a sequence of index pairs (i, j) satisfying the following conditions:

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

Algorithm 1 Curvature − Based Weight Adaptive Sampling

Require: Path points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M$, weight factors α and β, number of samples N
Ensure: Optimized sampling points $\mathbf{p}_{\text{sampled}}$

```
Initialize curvature array κ
for i = 2 to M − 1 do
    Compute v1 = pi − pi−1 and v2 = pi+1 − pi
    if ||v1|| == 0 or ||v2|| == 0 then
        κ[i − 1] ← 0
    else
        Compute curvature κ[i − 1] using Eq. (28)
    end if
end for
Normalize curvature using Eq. (29)
Compute weights w using Eq. (30)
Compute cumulative weights W(s) using Eq. (31)
Normalize cumulative weights using Eq. (32)
Initialize sampling points psampled
Define target weights Wtarget using Eq. (33)
for j = 1 to N do
    Find index i such that Wnorm(i) is closest to Wtarget[j]
    psampled[j] ← pi
end for
return psampled
```

1. $(1, 1) \in \pi$ and $(m, n) \in \pi$ (the alignment starts at the first points and ends at the last points of the curves).

2. The alignment is monotonic; that is, if $(i, j) \in \pi$, then $(i + 1, j)$, $(i, j + 1)$, or $(i + 1, j + 1)$ must also be in π.

The discrete Fréchet distance can be computed efficiently using a dynamic programming approach. Let $C[i, j]$ represent the discrete Fréchet distance between the sub-curves $\{p_1, \ldots, p_i\}$ and $\{q_1, \ldots, q_j\}$. The recursive formula for $C[i, j]$ is given by:

$$C[i,j] = \max\Big(\|p_i - q_j\|, \min\big(C[i-1,j],\ C[i,j-1],\ C[i-1,j-1]\big)\Big) \tag{35}$$

with the following base cases:

$$C[1,1] = \|p_1 - q_1\|, \tag{36}$$

$$C[i,1] = \max(C[i-1,1], \|p_i - q_1\|), \quad \text{for } i = 2, \ldots, m, \tag{37}$$

$$C[1,j] = \max(C[1,j-1], \|p_1 - q_j\|), \quad \text{for } j = 2, \ldots, n. \tag{38}$$

The final discrete Fréchet distance is given by:

$$d_F^d(P,Q) = C[m,n]. \tag{39}$$

The following pseudocode outlines the dynamic programming algorithm for computing the discrete Fréchet distance.

The time complexity of this algorithm is $O(m \cdot n)$, where m and n are the number of points in the curves P and Q, respectively. This makes it efficient for practical applications involving discrete curves.

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

Algorithm 2 Compute Discrete Fréchet Distance

Require: Two discrete curves $P = \{p_1, p_2, \ldots, p_m\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$
Ensure: The discrete Fréchet distance $d_F^d(P, Q)$

```
Initialize a 2D array C[1...m][1...n]
C[1][1] ← ||p_1 − q_1||
for i = 2 to m do
    C[i][1] ← max(C[i − 1][1], ||p_i − q_1||)
end for
for j = 2 to n do
    C[1][j] ← max(C[1][j − 1], ||p_1 − q_j||)
end for
for i = 2 to m do
    for j = 2 to n do
        C[i][j] ← max(||p_i − q_j||, min(C[i − 1][j], C[i][j − 1], C[i − 1][j − 1]))
    end for
end for
return C[m][n]
```

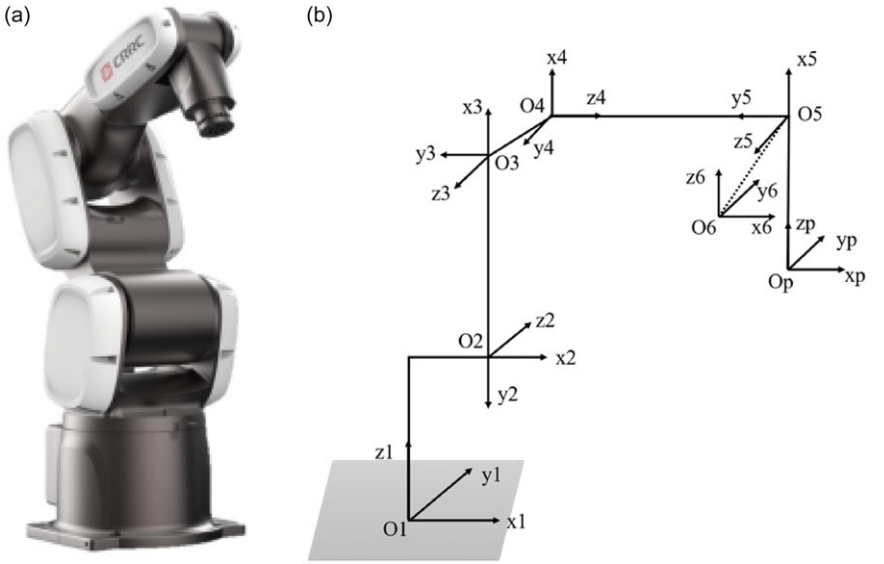


Figure 4. *Illustration of the robot manipulator and its corresponding DH coordinate system. (a) depicts the physical robot used in this study. (b) shows the abstract representation of the robot's kinematic structure using DH parameters.*

3. Experiments

3.1. Singularity consistent motion

We conducted validation experiments on a six-degree-of-freedom (6-DOF) serial manipulator jointly developed by our laboratory and an industrial manufacturer, whose kinematic layout follows the well-known PUMA-type architecture (shoulder–elbow–wrist with a spherical 3-axis wrist). The mechanical configuration is shown in Figure 4, and its Denavit–Hartenberg (DH) parameters are summarized in Table I. The control loop operates at a fixed frequency, and all planning and evaluation rely solely on the DH model and these bounds, independent of any vendor-specific features.

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

Table I. *Denavit–Hartenberg (DH) parameters of the robotic manipulator (in mm).*

Joint *i*	θ_i **(rad)**	d_i **(mm)**	a_i **(mm)**	α_i **(rad)**
1	θ_0	257.44	15	$-\frac{\pi}{2}$
2	$-\frac{\pi}{2}+\theta_1$	0	339.98	π
3	θ_2	0	35	$\frac{\pi}{2}$
4	θ_3	325.6	0	$-\frac{\pi}{2}$
5	$-\frac{\pi}{2}+\theta_4$	0	0	$-\frac{\pi}{2}$
6	θ_5	−175.94	0	0

We have designed a linear motion trajectory for the robotic arm, defined by a starting point p_1 and an endpoint p_2 in the Cartesian space:

$$p_1 : \{465.0000, 340.0000, -10.0000, -0.6283, 0, 0\},$$

$$p_2 : \{491.1970, 354.7178, 29.6084, 1.1903, -0.8956, 0.0806\}.$$

Along this trajectory, the path passes through a critical intermediate point, which corresponds to a singular configuration of the robotic arm. This singularity is characterized by the joint angles θ and the Cartesian coordinates p_s:

$$\theta : \{\pi/5, \pi/3, 0, \pi/3, \pi/2, \pi/6\},$$

$$p_s : \{478.0985, 347.3589, 9.8042, 0.2810, -0.4478, 0.0403\}.$$

The Jacobian matrix at the specified singularity is as follows:

$$J = \begin{bmatrix} -295.48 & -435.13 & 435.13 & 0 & 152.37 & 0 \\ 511.79 & -251.22 & 251.22 & 0 & -87.97 & 0 \\ 0 & -374.98 & 35.00 & 0 & 0 & 0 \\ 0.43 & -0.50 & 0.50 & 0 & -0.50 & 0 \\ 0.25 & 0.87 & -0.87 & 0 & -0.87 & 0 \\ 0.87 & 0.00 & 0.00 & -1 & 0.00 & 1 \end{bmatrix}$$

Near a singular configuration, the determinant of the Jacobian matrix approaches zero, indicating a loss of controllability. This singularity, specifically a wrist singularity, occurs when the fifth joint reaches 90 degrees. At this point, the manipulator's ability to exert forces or produce motion in certain directions is significantly reduced. Identifying such singularities is crucial for optimizing robot design and control.

At this specific point, the unit vector representing the design path parameter p is given by $S(p) = [0.7623, 0.0683, 1.5731, -0.7115, -0.8756, -0.8166]$. This vector defines the desired trajectory in the operational space. By integrating this parameter into the singularity-consistent framework, we can construct the augmented matrix as follows:

$$H = \begin{bmatrix} J & -S(p) \end{bmatrix}$$

$$= \begin{bmatrix} -295.48 & -435.13 & 435.13 & 0 & 152.37 & 0 & -0.76 \\ 511.79 & -251.22 & 251.22 & 0 & -87.97 & 0 & -0.07 \\ 0 & -374.98 & 35.00 & 0 & 0 & 0 & -1.57 \\ 0.43 & -0.50 & 0.50 & 0 & -0.50 & 0 & 0.71 \\ 0.25 & 0.87 & -0.87 & 0 & -0.87 & 0 & 0.88 \\ 0.87 & 0.00 & 0.00 & -1 & 0.00 & 1 & 0.82 \end{bmatrix}$$

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

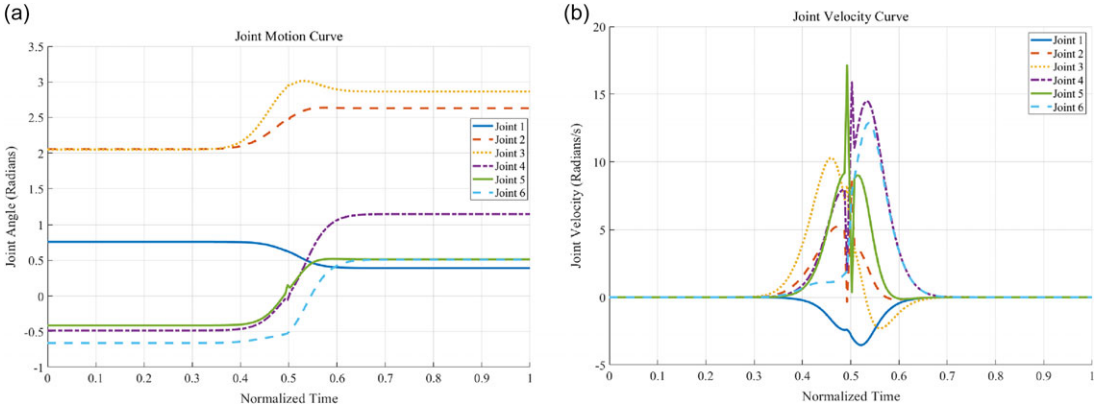


Figure 5. *Joint motion curves on the given singular path using the traditional kinematic method: (a) shows the joint position curves, and (b) shows the joint velocity curves.*

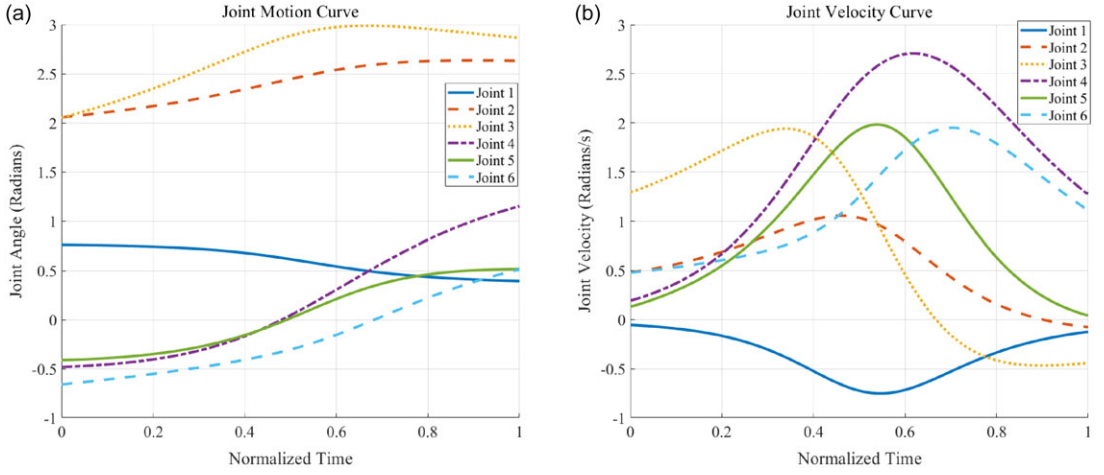


Figure 6. *Joint motion curves on the given singular path using the proposed method: (a) shows the joint position curves, and (b) shows the joint velocity curves.*

The introduction of the path vector ensures that the determinant of the augmented matrix H remains nonzero. Then, based on Eq. (15) and considering joint velocity limits, a velocity scaling factor b is calculated. This factor effectively suppresses excessive joint velocities, enabling the manipulator to traverse the singularity smoothly without abrupt motion.

To further highlight the effectiveness of the proposed method, we additionally implement a conventional kinematic approach under identical conditions, without the singularity-aware path vector and velocity scaling. As illustrated in Figure 5, the traditional method induces noticeable spikes in joint velocities when approaching the singularity, leading to unstable motion. In contrast, Figure 6 shows that our method maintains bounded velocities and smooth transitions through the singular configuration. This comparative analysis underscores the robustness and superiority of the proposed strategy.

3.2. Path curvature adaptive sampling

In this study, we designed and utilized a Bézier curve described by the following parametric equation:

$$\begin{aligned} B(t) = {} & (1-t)^4 P_0 + 4(1-t)^3 t P_1 \\ & + 6(1-t)^2 t^2 P_2 + 4(1-t) t^3 P_3 \\ & + t^4 P_4, \end{aligned} \tag{40}$$

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

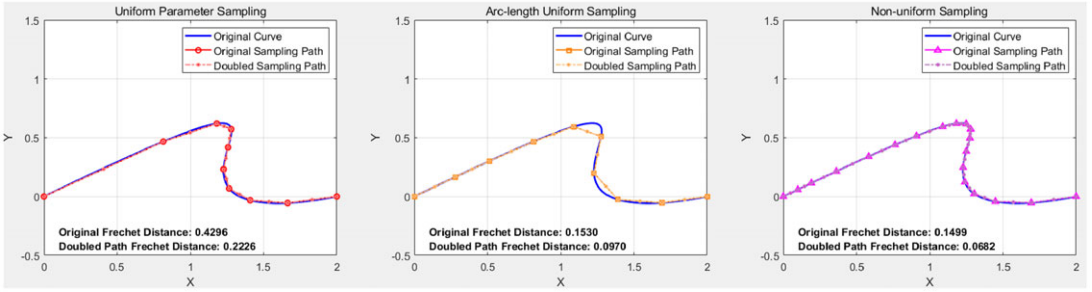


Figure 7. *Comparison of sampling methods. (Left) Uniform parameter sampling. (Center) Arc-length reparameterization uniform sampling. (Right) Adaptive sampling with curvature weighting.*

where the curve is defined by five control points, including a start point P_0, an end point P_4, and three intermediate control points P_1, P_2, and P_3. The specific values of these control points are as follows:

- $P_0 = [0, 0]$ (Start point),
- $P_1 = [2.5, 1.5]$
- $P_2 = [0.5, 0]$
- $P_3 = [1.2, -0.2]$
- $P_4 = [2, 0]$ (End point).

This curve above is a fourth-degree (quartic) Bézier curve. Its parametric equation provides a smooth and flexible path representation, with the intermediate control points P_1, P_2, and P_3 shaping the geometry and curvature. The parameter t varies continuously within the interval [0, 1], where $t = 0$ corresponds to the start point P_0 and $t = 1$ corresponds to the end point P_4. We use a Bézier curve because our method relies on arc-length–based smooth sampling over diverse curvature profiles; by adjusting control-point locations (and degree), one parametric curve yields rich – even non-monotonic – curvature while keeping path smoothness for stable derivatives and arc-length evaluation. Compared with more elaborate spline constructions, Bézier offers a simpler equation setup that still spans the required curvature regimes and is widely adopted in robotic/CNC trajectory planning, highlighting its practical engineering relevance.

The results of our experiments are depicted in Figure 7, which provides a comparative analysis of various sampling techniques.

Figure 7 illustrates the performance of three distinct sampling methods, evaluated using both the original and doubled path Fréchet distances. The original sampling path is generated by each method, and then the doubled sampling path is created by adding intermediate points to the original sampling path. The Fréchet distance between the original curve and the original sampling path reflects the initial approximation quality. The doubled path Fréchet distance, calculated between the original curve and the doubled sampling path, is used to further assess the geometric adherence of the original sampling path to the original curve. By comparing these two Fréchet distances, we can evaluate how well each sampling method captures the geometric essence of the original curve and how sensitive the approximation is to increased sampling density on the sampled path.

1. **Uniform Parameter Sampling:** As depicted in Figure 7 (left), this method maintains the general shape of the original curve but leads to significant variations in arc length between consecutive sample points. The original Fréchet distance (original curve to original sampling path) is 0.4296, and the doubled path Fréchet distance (original curve to doubled sampling path) is 0.2226.
2. **Arc-Length Reparameterization Uniform Sampling:** Figure 7 (center) demonstrates this technique, which achieves relatively uniform segment lengths. However, it introduces noticeable

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press



Figure 8. *Experimental platform of robot.*

geometric distortions, especially in regions with high curvature. The original Fréchet distance is 0.1530, and the doubled path Fréchet distance is 0.0970.

3. **Adaptive Sampling with Curvature Weighting (Non-uniform Sampling):** As illustrated in Figure 7 (right), this approach effectively balances shape fidelity and sampling density by allocating more sample points to high-curvature regions. The original Fréchet distance is 0.1499, and the doubled path Fréchet distance is 0.0682.

Analysis of the Fréchet distances, both original and doubled, provides a deeper understanding of the three sampling methods' performance. While a smaller original Fréchet distance indicates a better initial approximation of the original curve, the doubled path distance serves as a further validation of geometric adherence. The substantial reduction observed in uniform parameter sampling's doubled path distance highlights its initial limitations in capturing finer details, despite its simplicity. Arc-length reparameterization shows a better initial approximation, but the moderate reduction in its doubled path distance reveals limitations due to induced distortions. The weight adaptive sampling method consistently outperforms the others, exhibiting the lowest original and doubled path distances. This demonstrates its superior ability to accurately capture the original curve's geometry, a fact further reinforced by the doubled path analysis. The experiment underscores the importance of selecting a sampling method tailored to the specific application and desired accuracy, with weight adaptive sampling demonstrating the most robust performance in this case.

3.3. Physical verification

To validate the effectiveness of the proposed algorithm, we conducted physical experiments using a 6-DOF robotic manipulator (see Figure 8). The robot arm was securely mounted on a fixed base, with a fluorescent pen attached to its end-effector to serve as a writing instrument. A whiteboard was placed in front of the manipulator as the writing surface. Multiple trajectory patterns were designed to evaluate the robot's real-world motion tracking capabilities, including:

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

Table II. *Tracking errors (mean ± SD over N trials). Lower is better.*

Trajectory	Pos. RMSE (mm)	Vel. RMSE (mm/s)
Line–Arc	1.368×10^{-6}	0.00526
"2025" Digits	3.514×10^{-3}	0.197

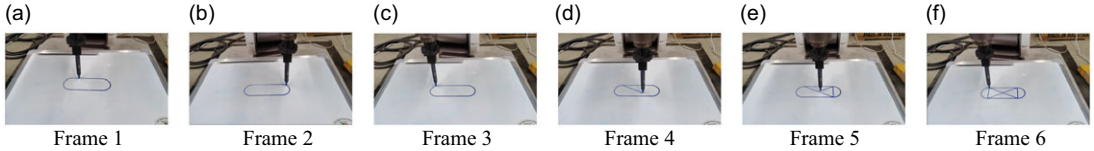


Figure 9. *Sequence of robot trajectory execution frames for the simple trajectory combination.*

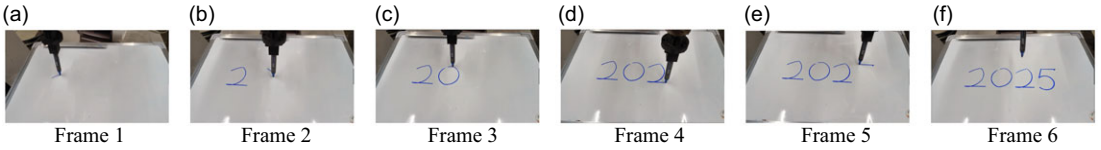


Figure 10. *Sequence of robot trajectory execution frames for the "2025" digit trajectory.*

1. Combinations of linear and circular arc trajectories, to verify the smoothness and stability of basic path following.
2. Complex "2025" digit trajectories, to assess the accuracy and adaptability of the manipulator when executing intricate paths.

Across trials, the manipulator executes the planned paths with stable motion, and the traced shapes closely match the references (Figures 9 and 10), indicating promising real-world feasibility. To complement qualitative observations, we report two standard errors between the measured and reference end-effector trajectories. Let $\mathbf{p}_t \in \mathbb{R}^3$ and $\mathbf{p}_t^*$ denote the measured and reference positions at synchronized timestamps $t = 1, \ldots, T$. The *positional RMSE* quantifies overall spatial accuracy:

$$\mathrm{RMSE}_{\mathrm{pos}} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \left\| \mathbf{p}_t - \mathbf{p}_t^* \right\|_2^2} \ [\mathrm{mm}].$$

To capture dynamic tracking fidelity, we compute velocities by differentiating lightly smoothed positions (e.g., Savitzky–Golay) to obtain $\dot{\mathbf{p}}_t$ and $\dot{\mathbf{p}}_t^*$. The *velocity RMSE* is then

$$\mathrm{RMSE}_{\mathrm{vel}} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \left\| \dot{\mathbf{p}}_t - \dot{\mathbf{p}}_t^* \right\|_2^2} \ [\mathrm{mm/s}].$$

All metrics are summarized as mean ± standard deviation over N repetitions per trajectory after timestamp alignment. The quantitative results in Table II further confirm these observations. For the line–arc trajectory, the mean position and velocity RMSEs of 1.368×10^{-6} mm and 5.26×10^{-3} mm/s demonstrate excellent tracking performance for smooth trajectories. For the more challenging "2025" digits, the errors increase to 3.514×10^{-3} mm and 0.197 mm/s, yet remain small relative to the overall path length and speed, demonstrating that the proposed algorithm maintains high accuracy even for intricate trajectories. The low standard deviations across repetitions suggest good repeatability of the physical experiments.

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

4. Conclusion

The study proposes a novel integrated framework for robotic arm path planning that addresses challenges of smooth motion control and precise geometric path tracking near singularities. The framework combines singularity-consistent kinematic representations for reconstructing motion patterns, arc-length reparameterization of Cartesian paths, and a curvature-weighted adaptive sampling strategy to dynamically adjust sampling intervals. This approach effectively balances joint velocity smoothness and geometric tracking accuracy.

Experimental validation compared three sampling methods – uniform parameter sampling, arc-length reparameterization, and the proposed weight adaptive sampling – using original and doubled Fréchet distances to quantitatively evaluate geometric adherence. Results show that the weight adaptive sampling consistently outperforms other methods by achieving the lowest Fréchet distances, indicating superior capability in capturing complex path geometries, especially near singular regions and high-curvature segments. Additionally, the singularity-consistent method demonstrated notably smoother transitions through singular configurations, mitigating abrupt joint motion changes and enhancing overall motion continuity.

Despite these significant improvements, several limitations remain. The computational complexity of curvature-weighted sampling can be substantial for long trajectories due to increased sample density, suggesting a need for optimization or more efficient curvature estimation techniques. Moreover, the current framework does not explicitly integrate obstacle avoidance, which is critical for applying the method in dynamic, cluttered environments. Robustness to external disturbances – such as unexpected forces, impacts, or payload changes – also requires further investigation and enhancement.

In conclusion, this study validates that integrating singularity-consistent motion control, curvature-adaptive sampling, and arc-length parameterization offers a robust and practical solution for robotic arm trajectory planning. The proposed framework significantly improves joint velocity continuity and geometric path adherence, ensuring smooth singularity transitions and precise tracking of complex Cartesian paths. Future work will focus on optimizing real-time computational performance, incorporating dynamic obstacle avoidance, and enhancing resilience to external disturbances, aiming to develop a more versatile and robust control strategy for robotic manipulators operating near singularities.

Author contributions. Xianyou Zhong conceived and designed the study. Xianyou Zhong and Mengjiao Shen developed the algorithms. Xianyou Zhong, Mengjiao Shen, and Xiao Lin collected and curated the data. Xianyou Zhong wrote the original draft. Chengju Liu and Qijun Chen provided resource support and writing guidance.

Financial support. This paper is supported by the National Natural Science Foundation of China (No. 62073245).

Competing interests. None.

Ethical approval. None.

References

[1] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME. J. Dyn. Sys., Meas., Control* **108**(3), 163–171 (1986).
[2] J. Manavalan and M. Howard. "Learning Singularity Avoidance," **In:** *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019) pp. 6849–6854.
[3] P. S. Donelan, "Singularity-theoretic methods in robot kinematics," *Robotica* **25**(6), 641–659 (2007).
[4] S. Caro, A. Semaan, D. Kanaan and P. Wenger, "Singularity analysis of the H4 robot using Grassmann–Cayley algebra," *Robotica* **30**(7), 1109–1118 (2012).
[5] W. Bu, "Closeness to singularities of robotic manipulators measured by characteristic angles," *Robotica* **34**(9), 2105–2115 (2016).
[6] W. Bu, "Closeness to singularities of manipulators based on geometric average normalized volume spanned by weighted screws," *Robotica* **35**(7), 1616–1626 (2017).
[7] S. Chiaverini, O. Egeland, "A Solution to the Singularity Problem for Six-Joint Manipulators," **In:** *IEEE International Conference on Robotics and Automation* (1990) pp. 644–649.

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press

[8] S. R. Buss, "Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Trans Robot. Autom.* **17**(1−19), 16 (2004).
[9] M. Kircánski, N. Kircánski, D. Lekovic and M. Vukobratovic, "An experimental study of resolved acceleration control of robots at singularities: Damped least-squares approach," *ASME J. Dyn. Sys., Meas., Control* **119**(1), 97–101 (1997).
[10] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. Syst., Man, Cybernet.* **16**(1), 93–101 (1986).
[11] S. R. Buss and J. S. Kim, "Selectively damped least squares for inverse kinematics," *J. Graph. Tools* **10**(3), 37–49 (2005).
[12] F. Zahedi, D. Chang and H. Lee, "User-adaptive variable damping control using Bayesian optimization to enhance physical human–robot interaction," *IEEE Robot. Autom. Lett.* **7**(2), 2724–2731 (2022).
[13] A. Maciejewski and C. Klein, "The singular value decomposition: Computation and applications to robotics," *Int. J. Robot. Res.* **8**(6), 63–79 (1989).
[14] M. Kirćanski, "Symbolic singular value decomposition for simple redundant manipulators and its application to robot control," *Int. J. Robot. Res.* **14**(4), 382–398 (1995).
[15] C. H. Choi, H. S. Park, S. H. Kim, H. J. Lee, J. K. Lee, J. S. Yoon and B. S. Park, "A Manipulator Workspace Generation Algorithm using a Singular Value Decomposition," **In:** *International Conference on Control, Automation and Systems* (2008) pp. 163–168.
[16] D. Nenchev and M. Uchiyama, "Singularity-Consistent Path Tracking: A Null Space Based Approach," **In:** *Proceedings of 1995 IEEE International Conference on Robotics and Automation* (1995) pp. 2482–2489.
[17] D. Nenchev, R. Okawa and H. Sone, "Task-space dynamics and motion/force control of fixed-base manipulators under reaction null-space-based redundancy resolution," *Robotica* **34**(12), 2860–2877 (2016).
[18] D. N. Nenchev, "Tracking Manipulator Trajectories with Ordinary Singularities: A Null-Space Based Approach," *Int. Confer. Control-Control* **2**, 1145–1147. 1994
[19] F. Vigoriti, F. Ruggiero, V. Lippiello and L. Villani, "Control of redundant robot arms with null-space compliance and singularity-free orientation representation," *Robot. Auton. Syst.* **100**, 186–193 (2018).
[20] M. Wojtyra and L. Wolihski, "On-Line Velocity Scaling Algorithm for Task Space Trajectories," **In:** *Symposium on Robot Design, Dynamics and Control* (2022) pp. 171–181.
[21] L. Petrović, F. Marić, I. Marković, J. Kelly and I. Petrović, "Trajectory Optimization with Geometry-Aware Singularity Avoidance for Robot Motion Planning," **In:** *2021 21st International Conference on Control, Automation and Systems (ICCAS)* (2021) pp. 1760–1765.
[22] P. Milenkovic, Z. Wang and J. Rodriguez, "Encountering singularities of a serial robot along continuous paths at high precision," *Mech. Mach. Theory* **181**, 105224 (2023).
[23] A. Bemporad, T. J. Tarn and N. Xi, "Predictive path parameterization for constrained robot control," *IEEE Trans. Control Syst. Tech.* **7**(6), 648–656 (1999).
[24] M. Kumon and N. Adachi, "Dynamic parameterization for path following control," *Asian J. Control* **3**(1), 27–34 (2001).
[25] Q. C. Pham and O. Stasse, "Time-optimal path parameterization for redundantly actuated robots: A numerical integration approach," *IEEE/ASME Trans. Mechatron.* **20**(6), 3257–3263 (2015).
[26] W. Suleiman, "On time parameterization of a robot path," *IFAC-Papers On Line* **48**(3), 52–57 (2015).
[27] D. Nenchev, Y. Tsumaki and M. Uchiyama, "Singularity-consistent parameterization of robot motion and control," *Int. J. Robot. Res.* **19**(2), 159–182 (2000).
[28] C. L. Chen and C. C. Peng, "Coordinate Transformation Based Contour Following Control for Robotic Systems," **In:** *Advances in Robot Manipulators* (IntechOpen, 2010).
[29] A. A. Khalate, G. Leena and G. Ray, "An adaptive fuzzy controller for trajectory tracking of robot manipulator," *Intell. Control Autom.* **2**(4), 364–370 (2011).
[30] C. Hu, Y. Chen and J. Wang, "Fuzzy observer-based transitional path-tracking control for autonomous vehicles," *IEEE Trans. Intell. Transp.* **22**(5), 3078–3088 (2020).
[31] J. Wang, J. Wu, T. Li and X. Liu, "Workspace and singularity analysis of a 3-DOF planar parallel manipulator with actuation redundancy," *Robotica* **27**(1), 51–57 (2009).
[32] A. V. Duka, "Neural network based inverse kinematics solution for trajectory tracking of a robotic arm," *Proc. Tech.* **12**, 20–27 (2014).
[33] A. Csiszar, J. Eilers and A. Verl, "On Solving the Inverse Kinematics Problem using Neural Networks," **In:** *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)* (2017) pp. 1–6.
[34] R. Sharma, S. Rana K.P. and V. Kumar, "Performance analysis of fractional order fuzzy PID controllers applied to a robotic manipulator," *Expert Syst. Appl.* **41**(9), 4274–4289 (2014).
[35] R. Mohammed, F. Bendary and K. Elserafi, "Trajectory tracking control for robot manipulator using fractional order-fuzzy-PID controller," *Int. J. Comput. Appl.* **134**(15), 22–29 (2016).
[36] T. Demjen, E.-C. Lovasz, M. Ceccarelli, C. Sticlaru, A.-M.-F. Lupuţi, A. Oarcea and D.-C. Silaghi-Perju, "Design of the five-bar linkage with singularity-free workspace," *Robotica* **41**(11), 3361–3379 (2023).

Cite this article: X. Zhong, M. Shen, X. Lin, C. Liu and Q. Chen (2025). "An adaptive path planning algorithm with singularity consistency for robotic arms", Robotica **43**, 4384–4401. https://doi.org/10.1017/S0263574725102890

https://doi.org/10.1017/S0263574725102890 Published online by Cambridge University Press