Spline-Based Approaches to Time-Optimal, Smooth, and High-Accuracy Trajectory Generation for Industrial Machines

（スプライン関数に基づく平滑性と高精度性を考慮した産業機械の最短時間動作軌道生成）

July, 2022

Doctor of Philosophy (Engineering)

Min Set Paing

ミン　セッ　パイン

Toyohashi University of Technology

Declaration of Authorship

I, Min Set Paing, declare that this thesis titled, "Spline-Based Approaches to Time-Optimal, Smooth, and High-Accuracy Trajectory Generation for Industrial Machines" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____

Date: July 2022

Abstract

Computer numerical control machines and industrial manipulators are extensively used in modern manufacturing industries to perform various tasks with complex profiles that require high-speed, high-accuracy, and repeatability. Tasks are represented as a geometric profile in the Cartesian workspace. Generally, complex profiles are separated into a series of linear segments by the computer-aided manufacturing system. To avoid discontinuous motions at the sharp corners of two linear segments, the machine has to completely stop the motion for each segment, which consumes energy and machining time. If the continuous feed rate is considered to transverse between segment trajectories, discontinuities of resulting trajectories induce mechanical wear and large tracking errors by the controllers.

To enhance productivity, the main objective is to complete the tasks in the minimum motion time. Consequently, research has explored to simultaneously improve the contradictory objectives of motion time, such as smoothness, consumed energy, and accuracy of the workpiece. Therefore, the Pareto optimal solution, which contributes a trade-off between contradictory objectives is getting attention by the decision makers nowadays. Due to the access limit of the original controllers and the complexity of the control structure, the advanced controllers are still difficult to implement for improving the desired objectives of industrial machines. Therefore, the optimal reference trajectory generation becomes a key driver for its simplicity. Most importantly, the reference motion trajectories must accurately represent the geometric path, be at least continuous in acceleration, and satisfy the kinematic and dynamic limits of the machines for all horizons.

In literature, the optimal control problems are solved by direct transcription methods, where the machine limits are considered as discrete constraints on grid points for calculation along the trajectory. As a drawback, constraint violations may occur in-between the determined

grid points. For complex paths with frequently changed derivatives, a large number of grids are typically required for constraint satisfaction; thus, the problem becomes computationally expensive, and the optimization solver may fail to find a feasible solution. Therefore, this process is not straightforward for users. This thesis presents several approaches for single and multi-objective smooth optimal trajectory generations for industrial machines by control parameterization of splines, guaranteeing kinematic constraints for the entire horizon regardless of the grid points. Two main approaches are analyzed in this study: decoupled, in which the optimal control problem is solved with two steps of geometric path planning and trajectory optimization, and coupled, in which the problem is solved in a single step of optimization. Literature review and preliminaries for this thesis are discussed in Chapter 1 and 2.

Chapter 3 and 4 provide the decoupled trajectory generations, assuming the geometric paths are predefined by a parametric curve or a set of via-points. The control parameterization is implemented in terms of B-splines to provide acceleration and jerk continuities. Chapter 3 presents the bi-objective optimization problem between the total motion and the jerk square integral of motion trajectories in the time domain. Guaranteed kinematic constraints are proposed based on the spline convex hulls and the maximum geometric derivatives along the path. The Pareto front comprising the significant trade-off solutions is explored by the combinations of normalized normal constraint and divide and conquer algorithms. The simulations and experimental results validate that the proposed method provides an approximately 3% faster and 21.57% lesser average axial tracking errors than the conventional linear reparameterization method. Chapter 4 discusses the time optimal control problem in the parameter domain using the nonlinear transformation of optimization variables. The time dependency is excluded, and the optimal motion time for each segment trajectory is independently achieved. The guaranteed kinematic constraints are determined based on the spline convex hulls that limit the maximum geometric derives between the locally affected intervals; therefore, it is more relevant to apply on complex geometric paths. Comparison with the jerk-limited time-optimal trajectory generation shows that the proposed method is more robust with the problem grid size and gives a smoother time-optimal trajectory, which reduces the average axial tracking errors by approximately 12%.

Chapter 5 presents coupled optimal trajectory generations by parameterization of B-splines. Time-optimal contour reshaping of the complex workpieces comprising straight-line, circle, and spline contour segments is discussed in the first part. The velocity and acceleration continuity between segment trajectories, fitting accuracy of the geometric path, and kinematic constraints

using spline convex hulls are considered as constraints in the problem formulation. The optimization results prove that the proposed method can represent various G-code segments by considering the trade-off between the time-optimality and fitting accuracy of the workpiece. The second part discusses the smooth trajectory generation for reduced impact motion. Motion optimization is considered for the smooth catching of a flying object with a similar velocity by minimizing the residual impact force between the object and the industrial machine. The optimization results confirm that the resulting trajectories are bounded by kinematic and workspace limits and satisfy the required velocity for the smooth catching operation. Finally, the conclusions for proposed spline-based decoupled and coupled optimal trajectory generations are drawn, and the expected future works are discussed in Chapter 6.

Acknowledgements

Firstly, I would like to express my great appreciation to my supervisor Prof. Naoki Uchiyama for his kindness, patience, and every support in this journey. Without his help, I am unable to come to Japan and achieve my Master Degree and PhD studies. I would like to show my gratitude to my teacher, Daw Khin Hnin Thu Zar for her kindness and guidance.

I would like to thank you very much to my thesis reviewers Prof. Kaiji Sato and Prof. Yohei Abe for the time and effort for reviewing the thesis and fruitful advice.

I would like to show my gratitude to the TUT scholarship for providing financially throughout these three years. The scholarship allows me exemption of tuition fees; therefore, I could emphasize more on my research studies. I take this opportunity to thank all the staff members from the Toyohashi University of Technology for their kindness and help.

Thank you very much for the members from the Systems Engineering Laboratory for their companionship and technical support for my research.

I had experienced up and down moments physically and mentally throughout this journey. I am very lucky to have the cheerful and supportive friends who listen to every single problem, especially Dr. Armando Tejeda Ochoa. Thank you very much from my heart.

Last but not the least, my great gratitude goes to my beloved Father, Mother, and Sister for believing me and giving me every support that I need from Myanmar.

Contents

List of Figures

List of Tables

Abbreviations

AC	Alternating Current
BOOP	Bi-objective Optimization Problem
CAD	Computer-aided Design
CAM	Computer-aided Manufacturing
CNC	Computer Numerical Control
CPU	Central Processing Unit
D&C	Divide and Conquer Algorithm
FDS	Feed Drive System
MOOP	Multi-objective Optimization Problem
NBI	Normal Boundary Intersection
NNC	Normalized Normal Constraint
NURBS	Non-uniform rational B-spline
OCP	Optimal Control Problem
PD	Proportional-derivative
PID	Proportional-integral-derivative

RAM	Random Access Memory
STOTG	Smooth Time-optimal Trajectory Generation
SQP	Sequential Quadratic Programming

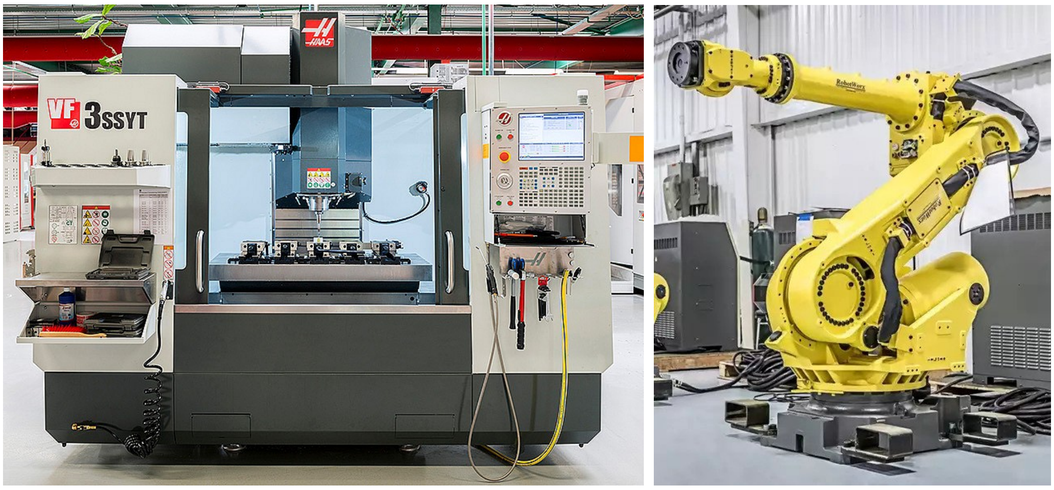
Chapter 1

Introduction and Literature Review

1.1 Motivation

In modern manufacturing industries, the development of science and technology enables high-speed and high-accuracy industrial machines such as CNC machines and robotic manipulators to perform a variety of tasks with complex profiles in the workspace (Fig. 1.1) [1–4]. Due to their repeatability and precise performance, these machines are extensively used to produce high-quality products around the clock.

This thesis mainly focuses on the optimal motion trajectory generation for industrial machines. For the productivity, the main objective is to accurately track a desired workpiece in the minimum motion time [5, 6]. The most conflicting objectives with the motion time in the literature are the trajectory smoothness [7] and accuracy of the workpiece [8]. Nowadays, the decision makers are interested about formulating the Pareto optimal solutions, which can give the trade-off level between the contradictory objectives. The conventional interpolator for complex workpieces consists of a series of linear segments provided by the Computer-aided Manufacturing (CAM) system. These trajectories are discontinuous in velocity and acceleration between the linear segments. The tracking performance of the reference trajectories is implemented by the CNC controllers, whose advanced strategies are designed to achieve the aforementioned objectives of industrial machines [9, 10]. However, these strategies are complicated to use in real applications and have been limited by the accessibility of original controllers. Besides,



(a) CNC milling machine. https://www.bkbprecision.com/

(b) Industrial robot manipulator. https://www.robots.com/

Fig. 1.1: Usage of CNC machines and robotic manipulators in manufacturing industries.

discontinuous and unbounded kinematic values of the reference trajectories may excite machine structures and induce large tracking errors by the control systems. Therefore, an optimal reference trajectory generation plays an important role in industrial manufacturing systems.

The motivation is that the optimal motion trajectories can be generated off-line considering the machine limits such as maximum velocity, acceleration, and jerk limits, and the quality requirements such as the accuracy of the workpiece. Thereafter, these optimal trajectories can be applied to the industrial system, without changing the original controllers for a high performance. Optimal trajectory generation techniques include coupled and decoupled approaches by direct or indirect optimization methods.

For the requirement of complex workpieces in automotive, aerospace, and biomedical parts, B-splines and Non-uniform rational B-spline (NURBS) are used as a standard form for the geometric modeling of curves and surfaces in Computer-aided Design (CAD) programs, and the motion primitives of industrial machines [11, 12]. The advantage of using splines over the conventional linear or circular interpolation is the faster motion time, flexibility, continuity in high derivative values, and the exact representation of complex geometric paths. This thesis presents a spline-based coupled and decoupled trajectory generations by a direct optimization approach. This study aims to address the time-optimal and Pareto optimal solutions with the development of guaranteed kinematically constrained structures for industrial machines.

Although the application of spline-based motion trajectories in this thesis is primarily for the bi-axial Feed Drive System (FDS), it can be extended to the multi-axial CNC machines and the robotic manipulators as well.

1.2 Literature review

1.2.1 Geometric path representation

For CNC machines, workpieces are designed by the CAD software and represented in a parametric form. The CAM system transforms the complex geometric profiles into a series of small segments for machining purposes. For robotic manipulators, the geometric path is given by a set of via-points determined by the high-level path planners (e.g., A* or hybrid A* algorithm [13]). The most used motion interpolators in common practice are the linear (G01) and the circular (G02/G03) segments. Using this conventional approach, the data transmission size becomes extremely large for accurate machining of complex geometric profiles [14]. Moreover, the machine has to accelerate and decelerate for each interpolated segment; therefore, the total motion time significantly increases. When the continuous feed rate is considered, discontinuities in velocity and acceleration occur between the segments, exciting the machine vibration and leading to the large tracking errors by the control systems [15].

Concerning the continuity of the path, there exists the geometric and parametric continuities [16, 17]. The paths provided by the CAD software are geometrically smooth, i.e., $\mathbb{G}^0$ for the position vector continuity, $\mathbb{G}^1$ for tangent vector continuity, and $\mathbb{G}^2$ for curvature vector continuity. Considering a complex geometric path, the classical approaches, cubic-splines [18], Bézier [19], and non-uniform rational B-splines (NURBS) [20] are widely used to represent geometric paths due to their continuity, flexibility, and the exact representation of complex profiles with a few variables. The simplified form of the NURBS with the weighting factor of one is the B-spline, which is a linear combination of piecewise polynomial functions, whose shape is determined by the vector of control points over the unified knot vector. Generally, two types of geometric fitting are considered: the interpolation (the curve or surface exactly passes through all the data points) and the approximation (the curve or surface passes through near the data points by minimizing the error between the data points) [11, 21, 22]. To accurately fit the complex geometric paths, tangent, normal, and curvature controls are included in the B-splines with the geometric algorithms [21, 23, 24]. In order to generate smooth trajectories, the geometric paths are required to be at least $\mathbb{G}^2$ continuous. Although the paths are geometric continuous, they may be parametrically discontinuous in velocity, acceleration, and jerk values unless the proper motion law or control parameterization is adopted for trajectory generation [25].

1.2.2 Curve motion interpolators

Given a geometric path represented as a parametric curve by B-splines or NURBS, the motion interpolators are used to obtain a commanded position for each sample time. The curve parameter is reparameterized in the time domain in terms of a specific motion law, which modifies the velocities, acceleration, and jerk vectors. In many cases, the motion law is considered as the constant scaling [25]. However, this method may not guarantee the smooth starting and ending of the trajectory i.e., the starting/ending velocity and acceleration are not zero.

Concerning a generic motion interpolation, Zhang and Greenway presented the curve interpolator with a desired trapezoidal feed rate based on the first-order Taylor series expansion [26]. Similarly, trapezoidal, exponential, and bell shapes profiles were used as the desired feed rate of the path, and the curve parameter is estimated for each sampling time by the second-order Taylor series expansion [14]. To reduce feed fluctuations, feed correction polynomial methods were proposed in [27, 28]. However, these methods require calculating the arc length of the geometric path to determine the motion time and the required velocity commands. Otherwise, arc length parameterization of the geometric path is required. The drawback is that inaccurate mapping between the arc displacement and the curve parameter may induce feed fluctuations, which produce high acceleration or jerk of the trajectories, exciting the machine structures and resulting in large tracking errors.

1.2.3 Optimal trajectory generation

In order to provide the drawbacks by conventional and curve motion interpolators, optimal trajectory generations were investigated in the literature. An optimal trajectory generation generally considers satisfying kinematic and dynamic limitations of machines and accuracy of the geometric path, and produces a reference trajectory with the desired objective. This procedure is naturally translated into the OCP by the control inputs, states, and outputs of the system, which satisfy the equality and inequality constraints along the horizon. The primitive trajectories are parameterized by smooth functions such as B-splines, Bezier, trapezoidal, s-curve, and polynomials. Because B-splines can accurately represent complex geometric paths with a high level of continuity, they have been extensively employed for control parameterization. There are two approaches of control parameterization for optimal trajectory generation: decoupled and coupled approaches [29, 30].

The decoupled approach separates the OCP into geometric path planning and optimal trajectory planning following the exact path. First, a geometric path is predefined as a parametric curve by one of the geometric algorithms discussed in Section 1.2.1, depending on the accuracy and correctness of the workpiece. After that, trajectory planning is considered to optimize the relation between the motion time and curve parameter to modify the velocity, acceleration, and jerk vectors with respect to the motion time. The advantage is that the geometric path does not change in the decoupled approach; therefore, it is suitable for industrial product designs that requires a precise presentation of geometric profiles, e.g., a CAD design.

On the other hand, the coupled approach solves both problems simultaneously. Therefore, the coupled approach requires more computation effort than the decoupled approach. Generally, tolerance of the geometric path is included as geometric constraints in an OCP, and contours of the geometric path are modified to obtain a desired objective. As a result, the geometric path changes in coupled approach. Concerning both decoupled and coupled approaches, mainly used objectives for OCPs in the literature are motion time, trajectory smoothness, accuracy of the geometric path, and consumed energy. Methods to solve the OCPs are reviewed in Section 1.2.4. The general flowchart for optimal trajectory generation of industrial machines is illustrated in Fig. 1.2.

1.2.3.1 Time-optimal trajectory generation

The main criterion for improving productivity is time optimality. Most importantly, the time-optimal trajectory has to satisfy the kinematic or dynamic limitations of the machines in a

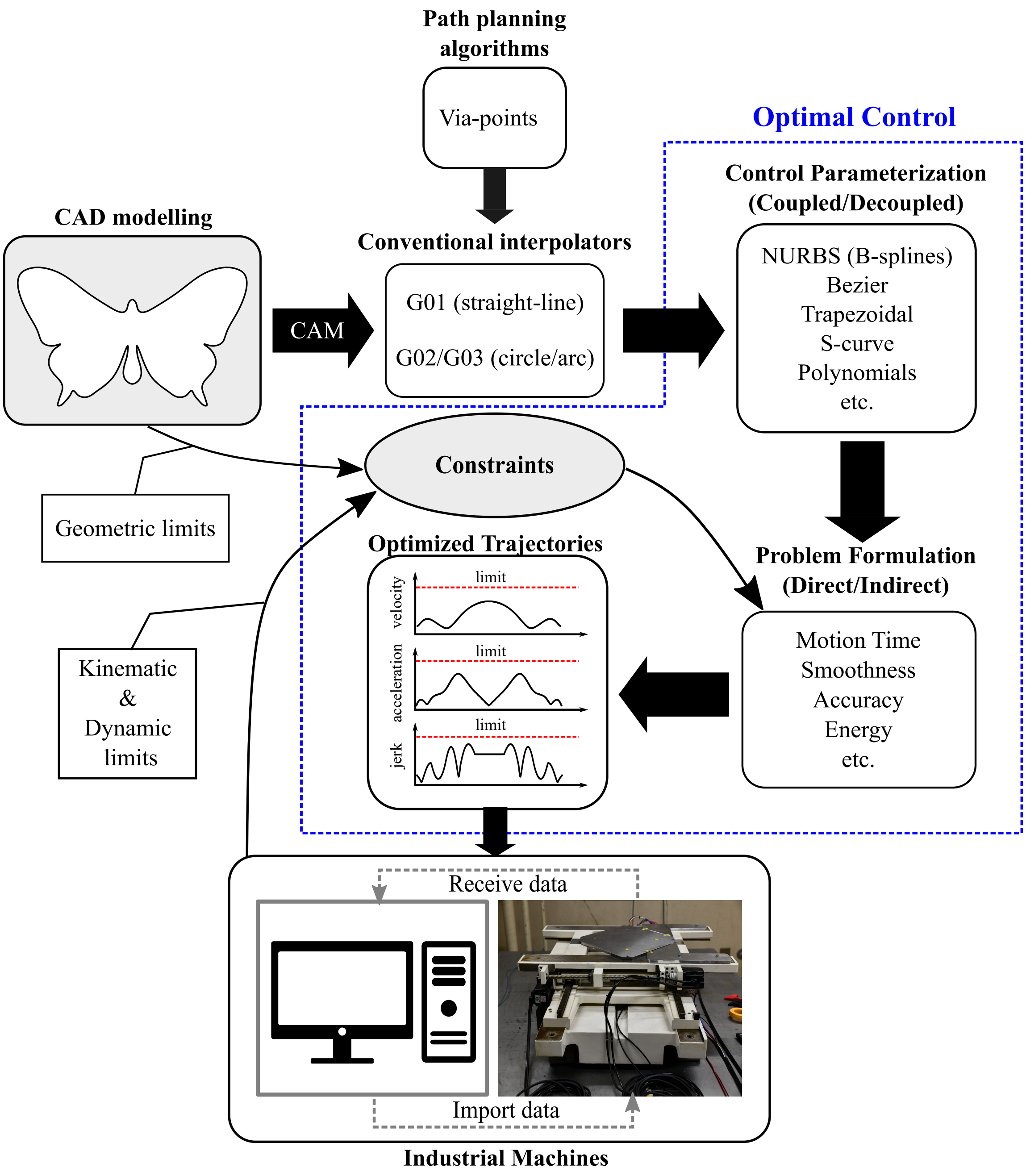


Fig. 1.2: Overview of the optimal trajectory generation for industrial machines

minimum motion time. Time-optimal trajectories were generated by the multiple switching curves along a specified path by the actuator torques/forces as the control inputs [5, 31]. The solutions of these algorithms consist of the bang-bang control, which made the discontinuity in acceleration or torques, exciting the high-frequency dynamics and producing significant vibrations [6, 32]. In order to address the discontinuous inputs, the modified cost function, e.g., time-energy, was presented [33, 34]. The original time-optimal trajectory generation only considers the constraints up to acceleration or torque limits in formulating the OCPs. The usage of these trajectories with a non-specialized controller may cause non-smooth movements. Using Parseval's theorem, Jamhour and André presented that constraining the jerk value improved the smoothness of the trajectory, which reduced machine vibrations [32]. For this reason, jerk constraints were considered in the smooth time-optimal trajectory generations [35, 36]. In [37], the combination of trajectory planning with cubic splines in the Cartesian space and B-splines in the joint space produced time-optimal and jerk-continuous trajectories. In [38], the trajectory was parameterized by the smooth cubic splines, and torque rates were considered in terms of pseudo-acceleration and jerk. The tracking errors caused by the conventional Proportional-integral-derivative (PID) controller were considered in time-optimal trajectory generation [39, 40] to improve the tracking accuracy.

1.2.3.2 Time-jerk optimal trajectory generation

The generation of trajectories with minimum jerk has been investigated because high jerk trajectories induce significant vibrations in actuators, which affect the tracking performance of control algorithms [41]. In [42], the Pontryagin's minimum principle was used to minimize two objectives: the maximum absolute value of the jerk and the time integral of the jerk square. In [43, 44], a cubic spline was used to parameterize the joint trajectory, and the interval analysis was used to globally minimize the maximum absolute value of the constant jerk.

The above-mentioned jerk minimized trajectories were generated within a predefined motion time without any kinematic constraints. The jerk minimization problem contradicts the time minimization because the trajectory reaches the extremities of the kinematic limits when time is kept minimum. Based on this concept, Gasparetto *et al.* formulated Bi-objective Optimization Problems (BOOPs) for robotic manipulators considering total motion time and integral of jerk square by the cubic splines [45] and by the fifth-order B-splines [7]. Time was not considered as a priori, and the kinematic limits were considered along the trajectory. The experiments for jerk-minimized trajectories were conducted in [46] and showed that algorithms with cubic splines [45] and B-splines [7] generated the lower mean jerk values and lower computation time than the global minimization algorithm [44]. In their study, the weighted sum method was used to formulate the BOOPs; however, a proper technique for choosing the weights for the trade-offs of contradictory objectives was not addressed.

1.2.3.3 Optimal trajectory generation with geometric constraints

Trajectory generations by decoupled approaches do not lead to a time-optimal movement [47]. Therefore, the OCPs were formulated considering the accuracy of the workpiece as a constraint or an objective function (coupled approaches). The corner smoothing methods were proposed for the smooth transitions between the linear segments [48–51]. In [52], the linear segments were expressed as two points jerk-limited acceleration profile and joined by the kinematic corner smoothing with interrupted acceleration. The OCP was formulated as a BOOP between the total motion time and accuracy at the corners. Similarly, the NURBS curve was used for the optimal cornering motion for linear segments [8]. These methods only addressed the optimal trajectory generations for linear segments by smooth cornering. In [30], Mercy *et al.* represented each a given linear (G01) or circle (G02/G03) segment by the cubic B-spline, and the time-optimal trajectory generation was formulated, assuring smooth transitions between each segment trajectory (continuous velocity and acceleration). Tolerance of each G-code segment was considered as a constraint in the OCP; however, a more complicated contour, e.g., a spline (G05), was not addressed. Therefore, the accuracy of the workpiece may be reduced without necessarily dividing the spline profile into multiple linear or circular segments.

1.2.4 Approaches to solve the optimal control problem

The OCP finds the inputs and the corresponding state trajectories from the initial state to the final state value by satisfying the state constraints by minimizing the performance measure (objective function) [53] (See Fig. 1.3) There are several approaches to solve the OCPs: the indirect and direct approaches [54, 55]. The indirect method solves the infinite OCP as the boundary value problem providing the necessary optimality conditions. Korayem *et al.* provided optimal trajectory planning strategies for mobile manipulators based on the Pontryagin's minimum principle [56–58]. Time-energy optimal trajectory generation with small tracking errors was proposed in [34]. A method for determining the maximum allowable load for mobile manipulators with redundancy constraints was presented in [59]. Indirect methods are numerically unstable and are difficult to implement and initialize.

On the other hand, the direct method discretizes the infinite OCP and converts it into the finite-dimensional nonlinear programming problem, which is solved by numerical optimization techniques. A direct method was applied to generate the optimal trajectory of rigid body systems with environmental contact [60]. The direct methods are based on the control parameterization of the state trajectory and are the most widespread and successfully used technique for constrained OCP in real-world applications [54, 61]. The direct approaches include single shooting, multiple shooting, and collocation techniques.

In the single shooting technique, the control inputs are discretized on the unit grid interval and rescaled into the time horizon. The state trajectory is regarded as an implicit function of the piecewise control input; therefore, the simulation of the state trajectory and optimization procedure goes sequentially one after another. Although it is simple, the unstable systems are difficult to implement with a single shooting [62]. The multiple shooting technique breaks down the problem into multiple single shootings and adds the state trajectory as the equality continuity constraints. The simulation of the state trajectory delivers when the optimization iterations end; therefore, it is considered as the simultaneous approach. Similar to multiple shooting, the collocation method uses the knots points, which are the connection of piecewise polynomials, and the state constraints are imposed at the collocation points [61–63]. The decision variables are the coefficients of polynomials, which change the value of the state and input at each knot. All these direct methods are successfully used to solve the OCPs for industrial machines.

However, all these methods require the discretization form of the constraints in order to avoid semi-infinite inequality constraints along the path. As a drawback, the constraints are not guaranteed to satisfy in-between the determined grid points along the horizon.

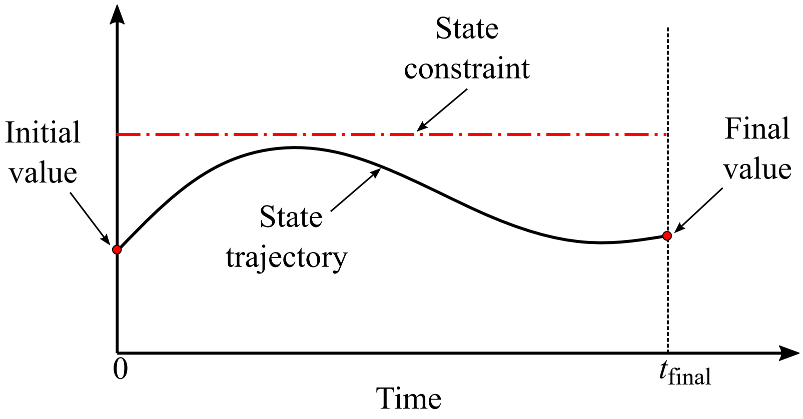


Fig. 1.3: Illustration of an OCP with the state constraint along the time horizon.

1.3 Thesis contributions and outline

1.3.1 Contributions

The time-optimal trajectories are the main necessity in modern manufacturing industries to improve productivity. At the same time, these time-optimal trajectories must follow the kinematic and dynamic capabilities of the industrial machines in order to prevent tool vibration and damage to the actuators induced by the discontinuous unbounded acceleration or jerk values. According to the industrial needs, the decision makers are interested about the Pareto optimal solutions, which gives the trade-off level between the total motion time and its contradictory objectives. In this thesis, spline-based single and multi-objective decoupled and coupled OCPs are proposed for optimal trajectory generation of industrial machines considering motion time, smoothness, and geometric accuracy. The dynamic constraints are not included in OCP formulation to avoid the high computational complexity. However, the kinematic constraint satisfaction largely affects the dynamic behavior of industrial machines (e.g., constraining jerk values improve the tracking accuracy of control systems). Therefore, these constraints can be considered as kinodynamic constraints. The main contributions are described as follows:

- The OCPs solved by the direct transcription method has been presented by setting discrete constraints on specific grids for calculations. The drawback is that when generating time-optimal trajectories of complex geometric paths whose derivative values are frequently changed over the horizon, the bang-bang behavior of the kinematic constraints may violate the machine limits, especially when the number of grids is small. Increasing the grids introduces additional variables and constraints in the OCP; therefore, it is computationally expensive. This thesis introduces a guaranteed satisfaction of kinematic limits for decoupled and coupled trajectory generation approaches regardless of the number of grids by exploiting spline properties.

- The usage of spline properties introduces the conservatism of kinematic constraints, where the kinematic values are distant from the maximum allowable limits. This study discusses a technique for reducing the conservatism by increasing the problem grids, along with the generated total motion time and the computation time of the trajectories. As an advantage, the trajectories move closer to the maximum limits for a faster motion.

- Several studies have considered time-optimal trajectories with jerk constraints to reduce machine tracking errors and tool vibrations. However, a few studies were considered to simultaneously address the contradictory nature of motion time and jerk. Therefore, in this study, the contradictory objectives of motion time and jerk square integral of the generated trajectory are formulated as a BOOP. The Pareto front comprising the trade-off solutions is explored by the combination of the NNC and D&C algorithms. Therefore, compared with the previous studies, the decision-maker can choose the specific trade-off level for manufacturing systems.

- The proposed optimal trajectories satisfy the kinematic limits for all horizons, whose results are verified not only by the simulation results but also by the experiments with a significant tracking error reduction compared to the related studies. Therefore, the proposed method is practical to use as a reference trajectory for improving the tracking accuracy of machines, where the original controllers are difficult to be changed.

- Several studies have considered the time optimality of the trajectory while reshaping the contours for the given G-codes provided by the CAM system, especially for the straight-line (G01) and circle (G02/G03) segments. Considering a complex geometric path, the workpiece accuracy might decrease without necessary dividing into multiple G01 or G02/03 segments. Therefore, this study presents a technique to simultaneously fit the complex workpieces comprising G01 and G02/G03 and spline(G05) segments while formulating the time-optimal trajectories.

1.3.2 Outline

The thesis can be organized into two parts, namely decoupled trajectory generation and coupled trajectory generation approaches. Chapter 3 and 4 focus on the formulation of decoupled OCPs, assuming the via-points or the geometric path is given. Chapter 5 presents coupled approaches, where the contour reshaping is considered simultaneously with time-optimal trajectory generation. This section describes the brief description of the following chapters as follows:

- Chapter 3 proposes a kinematically constrained reparameterization approach for industrial machines. The given via-points are firstly represented as a parametric curve by interpolation and approximation algorithms, and the sixth order B-spline reparameterization function is adopted for the nonlinear relationship between the curve parameter and time. The significant trade-offs between total motion time and smoothness of the trajectory (jerk square integral) are formulated as a BOOP, using the combination of the D&C algorithm with the NNC method, where each Pareto optimal solution is calculated by the Sequential Quadratic Programming (SQP). The kinematic constraints are proposed based on the spline convex hulls and the maximum geometric derivatives along the path. The effectiveness is discussed with the conventional linear reparameterization method with simulations and experiments with an industrial bi-axial FDS.

- Chapter 4 presents time-optimal trajectory generation of a predefined geometric path for CNC machines, provided by the CAD software. The OCP is transformed into the parameter domain to avoid time dependency, and the cubic B-spline is used to parameterize pseudo-states of the trajectory. The kinematic constraints in Chapter 3 are modified to check the spline convex hulls and the locally affected variable geometric derivatives along the path. Therefore, it is practical for complex geometric paths, where the derivatives are frequently changed along the parameter interval $[0, 1]$. The comparison is implemented with the jerk-constrained smooth time-optimal trajectory generation approach, which uses discrete constraints on grid points for calculation. The simulation and experimental results with an industrial bi-axial FDS show that the proposed method provides a smooth time-optimal trajectory while satisfying the kinematic constraints for all horizons regardless of the problem grids.

- Chapter 5 discusses two coupled approaches for optimal trajectory generation. The first approach is for the time-optimal trajectory generation and the contour reshaping of the geometric paths for CNC machines, consisting of the straight-line (G01), circles (G02/G03), and splines (G05) segments. The cubic B-spline is used to parameterize each segment trajectory, and the fitting error is proposed as the closest distance between the discretized via-points of the given G-code segments and the initialized B-spline. The smooth transition between segment trajectories is provided by velocity and acceleration continuity. The OCP is divided into several subproblems, which are solved by the simultaneous optimization method, namely a rolling horizon. The optimization results are demonstrated with a complex profile considering different limits on fitting accuracy.

  The second coupled trajectory generation is the smooth trajectory generation for industrial planar manipulators when receiving an object with a high impact force. This motion is investigated with the reduced impact catching of a flying object whose landing positions and velocities are accurately estimated by the vision system. The end-effector catches the object with a similar velocity, and the OCP is formulated to minimize the residual impact force after the catch. The optimization results are discussed with the three-degree of freedom planar manipulator.

- Chapter 6 summarizes the key findings of this thesis and clarifies the expected future works.

The chapters are organized as illustrated in Fig. 1.4.

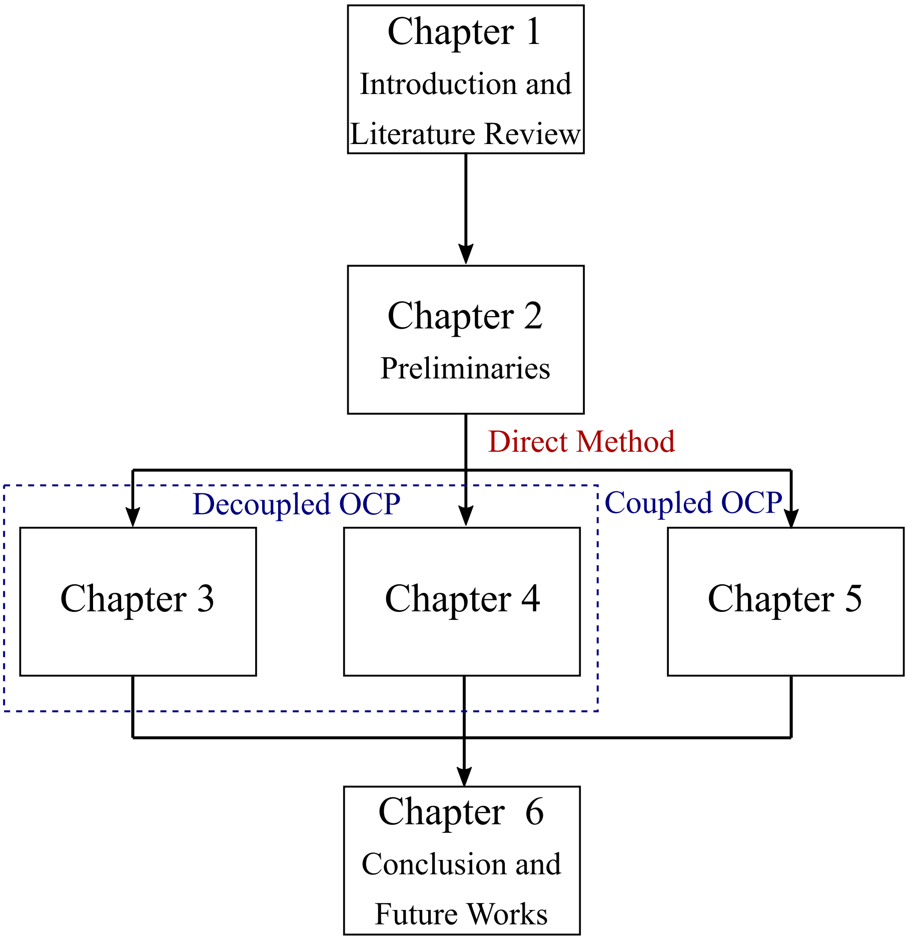


Fig. 1.4: Demonstration of thesis outline.

Chapter 2

Preliminaries

2.1 Spline parameterization of an optimal control problem

Splines are widely used for the control parameterization of motion trajectories due to their continuity, flexibility, and the exact representation of complex profiles with few variables [25, 64]. In this thesis, trajectories are parameterized in terms of B-splines, which provide at least continuity in the acceleration profile. In Chapter 3 and 4 (decoupled OCPs), the B-spline represents the parameter and pseudo-state values of the geometric path, whereas the B-spline represents the geometric positions in Chapter 5 (coupled OCPs). This section describes the definition of a spline-based trajectory and its important properties used in OCP formulation.

2.1.1 Spline trajectory

Concerning the decoupled and coupled OCPs, the motion trajectory can be parameterized in terms of B-splines for generating the smooth trajectory. The continuity of B-splines is dependent on the order of the curve k (degree, $p = k - 1$). Therefore, the k^{th} order B-spline curve consists of a piecewise linear combination of polynomial functions as follows:

$$\boldsymbol{s}(u) = \sum_{i=0}^{n} B_{i,k}(u) \;\; \boldsymbol{c}_i, \;\; u_{\min} \leqslant u \leqslant u_{\max}, \tag{2.1}$$

where $\boldsymbol{s}(u)$ represents the B-spline with parameter u, $u_{\min}$ and $u_{\min}$ are the user-defined minimum and maximum parameter values, respectively, $\boldsymbol{c}_j$ denotes the control point, n is the number of control points, and the knot vector of the B-spline consists of $(n + k + 1)$ knots that are k-times clamped at both ends. For simplicity, the knot points are not considered as optimization variables in this thesis. The non decreasing knot vector for (2.1) is defined as follows:

$$U = [\ \underbrace{u_{\min},\ ...,\ u_{\min}}_{k\text{-times}},\ u_k, u_{k+1}, ..., u_n,\ \underbrace{u_{\max},\ ...,\ u_{\max}}_{k\text{-times}}\ ], \tag{2.2}$$

Depending on the distribution of the knot vector in (2.2), the basis function is defined by the Cox-de Boor recursion formula [64] as follows:

$$\begin{aligned} B_{j,1}(u) &= \begin{cases} 1\ , & \text{for} \quad u_j \leqslant u < u_{j+1}. \\ 0\ , & \text{otherwise.} \end{cases} \\ B_{j,k}(u) &= \frac{(u - u_j)\, B_{j,k-1}(u)}{(u_{j+k-1} - u_j)} + \frac{(u_{j+k} - u)\, B_{j+1,k-1}(u)}{(u_{j+k} - u_{j+1})}. \end{aligned} \tag{2.3}$$

An illustration of B-spline parameterization is provided in Fig. 2.1. The r^{th} derivative of the B-spline with respect to parameter u is determined as follows:

$$\boldsymbol{s}^r(u) = \sum_{i=0}^{n} B^r_{i,k}(u)\ \ \boldsymbol{c}_i,\ \ r = \{1, 2, ..., k-1\}, \tag{2.4}$$

with its r^{th} derivative of the basis function given by

$$B^r_{j,k}(u) = (k-1)\left[\frac{B^{r-1}_{j,k-1}(u)}{u_{j+k-1} - u_j} - \frac{B^{r-1}_{j+1,k-1}(u)}{u_{j+k} - u_{j+1}}\right]. \tag{2.5}$$

From (2.5) and (2.4), the first derivative of a B-spline can be rewritten by

$$\boldsymbol{s}'(u) = \sum_{i=0}^{n-1} B_{i,k-1}(u)\, \boldsymbol{c}', \tag{2.6}$$

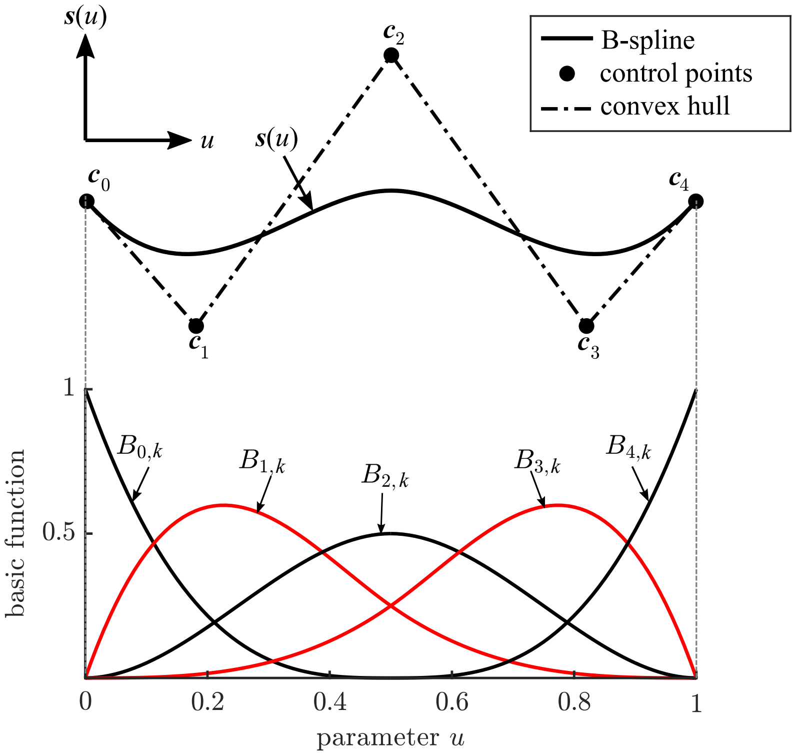


Fig. 2.1: Parameterization of the geometric path by a B-spline ($k = 4$) with the control points and basis functions in parameter $u \in [0, 1]$.

with

$$\boldsymbol{c}_i' = \frac{(k-1)}{u_{j+k} - u_{j+1}} \, (\boldsymbol{c}_{i+1} - \boldsymbol{c}_i), i = \{0, 1, ..., n-1\}, \tag{2.7}$$

where c_i' is the control point of the first derivative of the B-spline function, which corresponds to the knot vector as follows:

$$U' = [\ \underbrace{u_{\min},\ \ ...,\ \ u_{\min},}_{(k-1)\text{-times}}\ u_k, u_{k+1}, ..., u_n,\ \underbrace{u_{\max},\ \ ...,\ \ u_{\max}}_{(k-1)\text{-times}}\ ], \tag{2.8}$$

Similarly, the higher order derivatives of the B-spline curve can be determined in a recursive manner as follows:

$$\boldsymbol{s}^r = \sum_{i=0}^{n-r} B_{i,k-r}(u) \ \boldsymbol{c}_i^r, \tag{2.9}$$

with

$$\boldsymbol{c}_i^r = \begin{cases} \boldsymbol{c}_i \ , & \text{for} \quad k = 0, \\ \dfrac{k - r}{(u_{i+k} - u_{i+r})} \left(\boldsymbol{c}_{i+1}^{r-1} - \boldsymbol{c}_i^{r-1}\right) \ , & \text{for} \quad k > 0. \end{cases} \tag{2.10}$$

and the corresponding knot vector as follows:

$$U^r = [\ \underbrace{u_{\min},\ ...,\ u_{\min}}_{(k-r)\text{-times}},\ u_k, u_{k+1}, ..., u_n,\ \underbrace{u_{\max},\ ...,\ u_{\max}}_{(k-r)\text{-times}}\ ], \tag{2.11}$$

2.1.2 Spline properties

The most powerful property of B-splines used for trajectory generation is the convex hull property [11, 65]. The convex hull property states that the spline function is always contained in the convex hull, determined by its control points. According to the local modification scheme, the change of the control point c_i will only affect the spline function on the interval $[u_i, u_{i+k+1})$. Therefore, these properties are used to satisfy the semi-infinite constraints for optimal trajectory generations. For example, the semi-infinite constraint

$$\boldsymbol{s}_{\min} \leqslant \boldsymbol{s}(u) \leqslant \boldsymbol{s}_{\max}, \tag{2.12}$$

is guaranteed if the following finite set of constraints are satisfied:

$$\boldsymbol{s}_{\min} \leqslant \boldsymbol{c}_i \leqslant \boldsymbol{s}_{\max}, \quad i = \{0, 1, 2, ..., n\}. \tag{2.13}$$

where $\boldsymbol{s}_{\min}$ and $\boldsymbol{s}_{\max}$ are the maximum and minimum limits of the spline function, respectively. Compared to the classical optimization approaches, which use the discrete constraints on grid points for calculation, constraints using the convex hull property give a faster computation time with a guaranteed solution for constraint satisfaction for all horizons (See Fig. 2.2). A similar concept is applied to the derivatives of the splines. Therefore, the semi-infinite constraint for the r^{th} derivative of B-spline

$$\boldsymbol{s}_{\min}^r \leqslant \boldsymbol{s}^r(u) \leqslant \boldsymbol{s}_{\max}^r \tag{2.14}$$

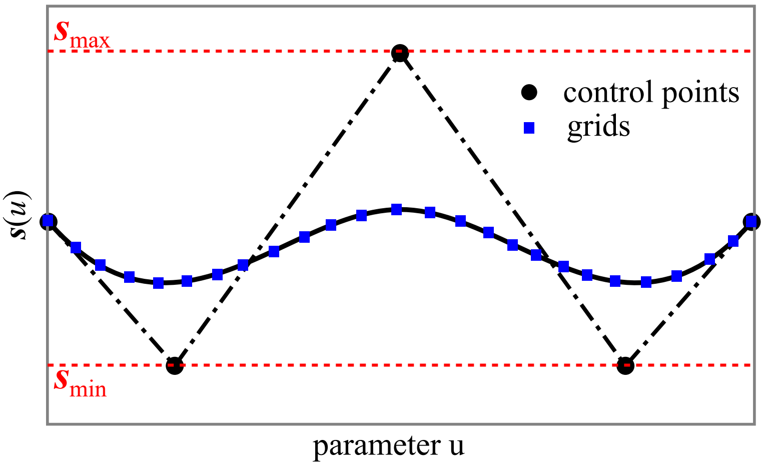


Fig. 2.2: Comparison of discrete constraints at grid points and control points along the trajectory.

is satisfied if the following condition by its control points are met:

$$\boldsymbol{s}^r_{\text{min}} \leqslant \boldsymbol{c}^r_i \leqslant \boldsymbol{s}^r_{\text{max}}, \quad i = \{0, 1, 2, ..., n - r\}. \tag{2.15}$$

where $\boldsymbol{s}^r_{\text{min}}$ and $\boldsymbol{s}^r_{\text{max}}$ are the minimum and maximum limits on the r^{th} derivative of the B-spline, respectively. Eq. (2.14) is generally used for kinematic constraints satisfaction (e.g., velocity, acceleration, and jerk) of the coupled OCPs. One of the drawbacks of using constraints by the spline convex hull property introduces some conservatism (i.e., the actual values are far from the maximum allowable limits). This conservatism can be reduced by the degree elevation or knot insertion algorithms in the literature [66, 67]. However, the constraints for decoupled approaches are still challenging because the generated velocity, acceleration, and jerk values depend on both the predefined path's derivatives and the control spline function itself. This thesis focuses on the kinematic constraint satisfaction on decoupled OCPs in Chapter 3 and 4.

2.2 Geometric curve fitting of given via-points

The geometric paths are given as a set of via-points in the Cartesian space, which are usually provided by the high-level path planning algorithms (e.g., A* algorithm or hybrid A* algorithms considering the obstacle avoidance scheme) or by discretization of a CAD profile by

CAM systems. In decoupled approaches of this thesis, global interpolation or approximation algorithms are used to predefine the given via-points as a parametric curve as below.

2.2.1 Global interpolation

Given a set of r via-points $\{\boldsymbol{d}_0, \boldsymbol{d}_1, ..., \boldsymbol{d}_r\}$, the global interpolation algorithm [11] calculates the control points which makes the B-spline curve passing through all these via-points. In this case, we assume that the parameter u lies within the range $[0, 1]$. The number of control points of the fitted B-spline curve equals the number of via-points, i.e., $n = r$. Using (2.1), $(n + 1)$ number of linear equations are defined at assigned parameter values $\tilde{u}_i$, $i = \{0, 1, ..., n\}$ as follows:

$$\boldsymbol{s}(\tilde{u}_i) = \sum_{i=0}^{n} B_{i,k}(\tilde{u}_i) \;\; \boldsymbol{c}_i = \boldsymbol{d}_i, \tag{2.16}$$

where $\boldsymbol{s}$ represents the three-dimensional geometric position. In matrix form, it can be represented for each m^{th} direction by the following:

$$\begin{aligned}
&\boldsymbol{B}\,\boldsymbol{C}_m = \boldsymbol{D}_m,\\
&\boldsymbol{B} = \begin{bmatrix} B_{0,k}(\tilde{u}_0) & B_{1,k}(\tilde{u}_0) & \dots & B_{n,k}(\tilde{u}_0) \\ B_{0,k}(\tilde{u}_1) & B_{1,k}(\tilde{u}_1) & \dots & B_{n,k}(\tilde{u}_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_{0,k}(\tilde{u}_n) & B_{1,k}(\tilde{u}_n) & \dots & B_{n,k}(\tilde{u}_n) \end{bmatrix}, \boldsymbol{C}_m = \begin{bmatrix} c_{0,m} \\ c_{1,m} \\ \vdots \\ c_{n,m} \end{bmatrix}, \boldsymbol{D}_m = \begin{bmatrix} d_{0,m} \\ d_{1,m} \\ \vdots \\ d_{n,m} \end{bmatrix},
\end{aligned} \tag{2.17}$$

The choice of $\tilde{u}_i$ effects the shape of the geometric path [68]. Here, the centripetal method is used to choose the assigned parameter values as follows:

$$\begin{aligned}
&\tilde{u}_0 = 0, \quad \tilde{u}_n = 1,\\
&\tilde{u}_i = \tilde{u}_{i+1} + \frac{\sqrt{|\boldsymbol{d}_i - \boldsymbol{d}_{i-1}|}}{D}, \;\; i = \{1, 2, ..., n-1\},
\end{aligned} \tag{2.18}$$

with

$$D = \sum_{i=0}^{n} \sqrt{|\boldsymbol{d}_i - \boldsymbol{d}_{i-1}|} \tag{2.19}$$

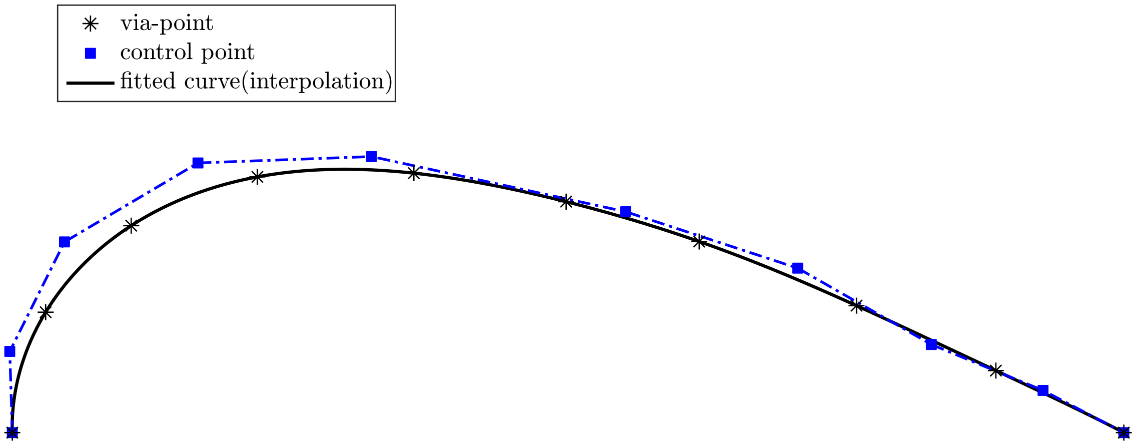


Fig. 2.3: Global interpolation of given 10 via-points by a B-spline with 10 control points.

To define the basis functions in (2.17), the knot vector U must be determined. Uniform distribution of the knot vector can be selected; however, this may generate a singular matrix $\boldsymbol{B}$ in (2.17) with the use of assigned parameters $\tilde{u}_i$ in (2.18). Therefore, it is recommended to choose the knot vector by the averaging method as follows:

$$\begin{aligned} &u_0 = ... = u_{k-1} = 0, \quad u_{n+2} = ... = u_{n+k+1} = 1, \\ &u_{j+k-1} = \frac{1}{(k-1)} \sum_{i=j}^{j+k-2} \tilde{u}_i, \;\; j = \{1, 2, ..., n-k+1\}. \end{aligned} \tag{2.20}$$

Thereafter, the control points for each axis can be calculated by solving $\boldsymbol{C}_m = \boldsymbol{B}^{-1}\,\boldsymbol{D}_m$. An illustration of the global interpolation of a given via-points is given in Fig. 2.3. When the data size of given via-points is very large, the interpolation of each via-point may occur the fluctuation of the fitting curve; hence, in this case, it is recommended to use an approximation of given via-points instead.

2.2.2 Global approximation

The global approximation algorithm finds a B-spline curve, which passes exactly through the first and last via-points and near the remaining via-points by minimizing the error distance between them. [11, 69, 70] The given set of via-points $\{\boldsymbol{d}_0, \boldsymbol{d}_1, ..., \boldsymbol{d}_m\}$ are approximated by k^{th}

order B-spline with $(n+1)$ control points, where $n < m$ as follows:

$$\boldsymbol{s}(u) = \sum_{i=0}^{n} B_{i,k}(u)\;\; \boldsymbol{c}_i,\;\; 0 \leqslant u \leqslant 1. \tag{2.21}$$

In this case, the first and last control points are simply computed as $\boldsymbol{c}_0 = \boldsymbol{d}_0$ and $\boldsymbol{c}_n = \boldsymbol{d}_m$, and the remaining control points are computed in the sense of least squares as follows:

$$\sum_{i=1}^{m-1} \;|\boldsymbol{d}_i - \boldsymbol{s}(\tilde{u}_i)|^2, \tag{2.22}$$

where $\tilde{u}_i$ is the assigned parameter according to (2.18) and (2.19). Following the derivation in [11], the matrix form can be represented for the m^{th} direction as follows:

$$\begin{aligned}
&(\boldsymbol{B}^T\; \boldsymbol{B})\, \boldsymbol{C}_m = \boldsymbol{Q}_m \\
&\boldsymbol{B} = \begin{bmatrix} B_{1,k}(\tilde{u}_1) & B_{2,k}(\tilde{u}_1) & \cdots & B_{n-1,k}(\tilde{u}_1) \\ B_{1,k}(\tilde{u}_2) & B_{2,k}(\tilde{u}_2) & \cdots & B_{n-1,k}(\tilde{u}_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_{1,k}(\tilde{u}_{r-1}) & B_{2,k}(\tilde{u}_{r-1}) & \cdots & B_{n-1,k}(\tilde{u}_{r-1}) \end{bmatrix}, \boldsymbol{C}_m = \begin{bmatrix} c_{0,m} \\ c_{1,m} \\ \vdots \\ c_{r-1,m} \end{bmatrix}, \\
&\boldsymbol{Q}_m = \begin{bmatrix} B_{1,k}(\tilde{u}_1)\, q_{1,m} \;+\; B_{1,k}(\tilde{u}_2)\, q_{2,m} \;+\; \cdots \;+ B_{1,k}(\tilde{u}_{r-1})\, q_{r-1,m} \\ B_{2,k}(\tilde{u}_1)\, q_{1,m} \;+\; B_{2,k}(\tilde{u}_2)\, q_{2,m} \;+\; \cdots \;+ B_{2,k}(\tilde{u}_{r-1})\, q_{r-1,m} \\ \vdots \\ B_{n-1,k}(\tilde{u}_1)\, q_{1,m} \;+\; B_{n-1,k}(\tilde{u}_2)\, q_{2,m} \;+\; \cdots \;+ B_{n-1,k}(\tilde{u}_{r-1})\, q_{r-1,m} \end{bmatrix} \\
&q_{i,m} = d_{i,m} - B_{0,k}(\tilde{u}_i)\, d_{0,m} - B_{n,k}(\tilde{u}_i)\, d_{r,m}, i = \{1, 2, \cdots, r-1\}.
\end{aligned} \tag{2.23}$$

To guarantee every knot span includes at least one $\tilde{u}_i$ which makes the matrix $(\boldsymbol{B}^T\; \boldsymbol{B})$ positive definite, the internal knots are defined as follows:

$$\begin{aligned}
&u_{k+j-1} = (1-\alpha)\, \tilde{u}_{i-1} + \alpha\, \tilde{u}_{i-1},\;\; j = \{1, 2, \cdots, n-k+1\}, \\
&i = \text{int}(jd),\;\; d = \frac{r+1}{n-k+2}\;\; \alpha = jd - 1.
\end{aligned} \tag{2.24}$$

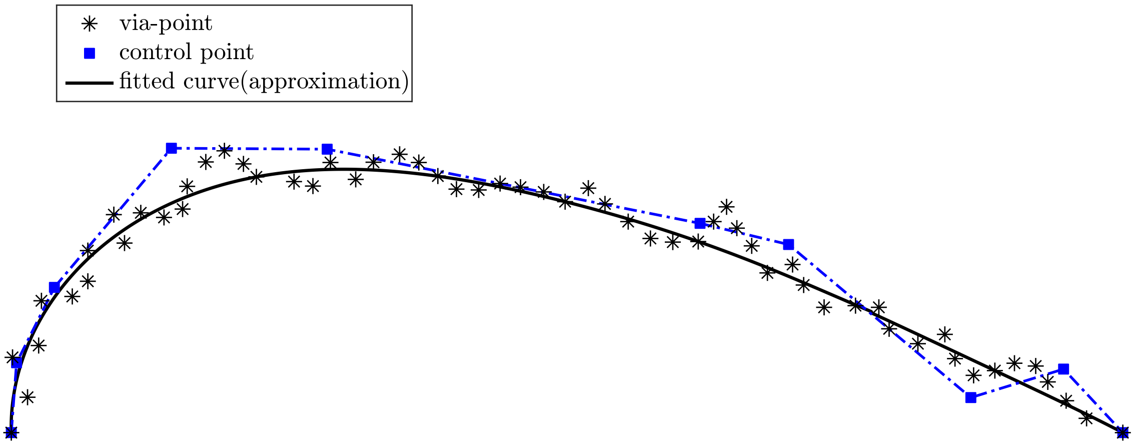


Fig. 2.4: Global approximation of given 60 via-points by a B-spline with 10 control points.

Thereafter, the control points are determined by calculating $\boldsymbol{C}_m = (\boldsymbol{B}^T\ \boldsymbol{B})^{-1}\,\boldsymbol{Q}_m$. Approximation of a given via-points by global approximation algorithm is illustrated in Fig. 2.4.

2.3 The optimal control problem formulation

In this thesis, the OCPs are formulated as a single-objective or a Multi-objective Optimization Problems (MOOPs). In MOOPs, the generated solution must be Pareto-optimal, where there exists no other feasible solution that improves the value of at least one objective without deteriorating the other objectives [71, 72]. The set consisting of Pareto-optimal solutions is called the Pareto front. The general MOOP is defined as follows:

$$\min_{\boldsymbol{\mu}}\{F_1(\boldsymbol{\mu}), F_2(\boldsymbol{\mu}), ..., F_n(\boldsymbol{\mu})\},\ \ n \leqslant 2. \tag{2.25}$$

subject to

$$\boldsymbol{h}(\boldsymbol{\mu}) = 0, \tag{2.26}$$

$$\boldsymbol{g}(\boldsymbol{\mu}) \leqslant 0, \tag{2.27}$$

$$\boldsymbol{\mu}_{\text{min}} \leqslant \boldsymbol{\mu} \leqslant \boldsymbol{\mu}_{\text{max}}, \tag{2.28}$$

where, $\boldsymbol{\mu}$ is the optimization variable vector, $\boldsymbol{h}(\boldsymbol{\mu})$ is the equality constraint vector, $\boldsymbol{g}(\boldsymbol{\mu})$ is the inequality constraint vector, and $\boldsymbol{\mu}_{\text{min}}$ and $\boldsymbol{\mu}_{\text{max}}$ are the upper and lower bounds of $\boldsymbol{\mu}$,

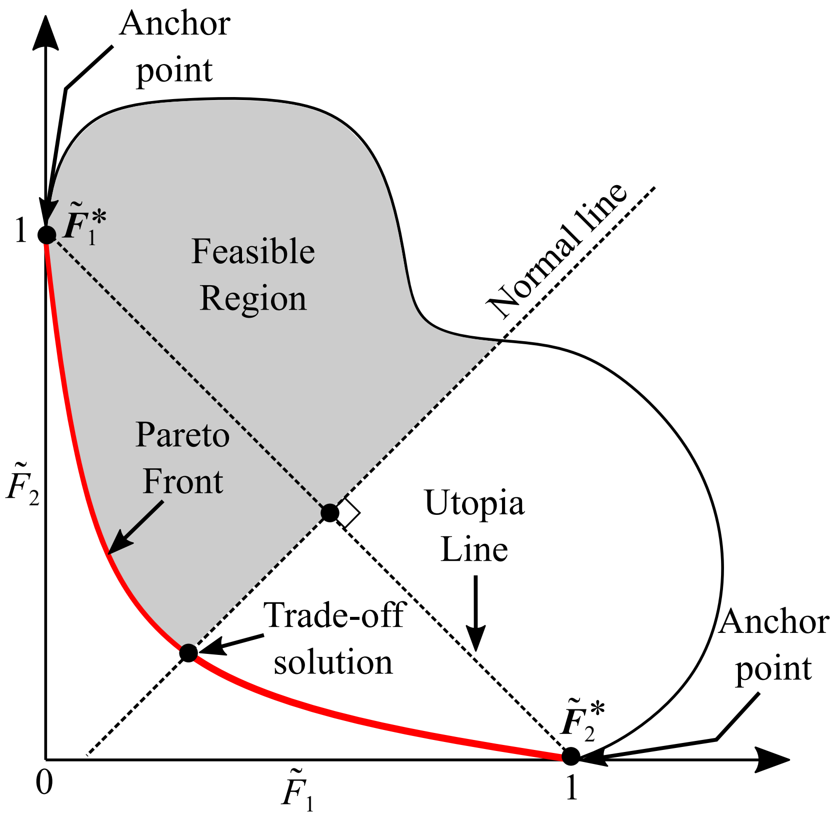


Fig. 2.5: Demonstration of the NNC method for bi-objective case.

respectively. There are several approaches for obtaining the optimal solutions of MOOP. The widely used method is the weighted sum method, where each objective function is multiplied by a weighting factor, then sums up all to form an aggregated scalar objective function [71, 73, 74]. However, this method does not give a well-distributed set of Pareto solutions and cannot generate solutions in concave Pareto regions. This section describes the process of revealing the Pareto front between contradictory objectives by the combination of NNC method and D&C algorithm.

2.3.1 Normalized normal constraint method

The NNC is a method for generating the Pareto front by minimizing a normalized objective and incorporating the remaining objectives as additional constraints [75]. Considering a BOOP ($n = 2$), the anchor points (the optimal vertices) are determined by minimizing each of the objective function, independently as follows:

$$f_{1,\min} = \min_{\boldsymbol{\mu}} \; J_1(\boldsymbol{\mu}), \quad f_{2,\min} = \min_{\boldsymbol{\mu}} \; J_2(\boldsymbol{\mu}) \tag{2.29}$$

subject to (2.26)-(2.28). Thereafter, the anchor points are mapped into the normalized objective space as $\tilde{\boldsymbol{F}}_1^*(0,1)$ and $\tilde{\boldsymbol{F}}_2^*(1,0)$ shown in Fig. 2.5. The line joining the two anchor points is called the Utopia line. The objective functions are normalized as follows:

$$\tilde{F}_1 = \frac{F_1 - f_{1,\min}}{f_{1,\max} - f_{1,\min}}, \;\; \tilde{F}_2 = \frac{F_2 - f_{2,\min}}{f_{2,\max} - f_{2,\min}}, \tag{2.30}$$

with

$$\begin{aligned} f_{1,\max} = t_{\max}, \qquad f_{2,\max} &= F_2(\boldsymbol{\mu}_{1,\min}), \\ \boldsymbol{\mu}_{1,\min} &= \arg\min_{\boldsymbol{\mu}} F_1. \end{aligned} \tag{2.31}$$

Therefore, the bi-objective optimization problem is reformulated by the NNC as follows:

$$\min_{\boldsymbol{\mu}} \tilde{F}_2, \tag{2.32}$$

subject to (2.26)-(2.28), and:

$$\boldsymbol{n}^T[\boldsymbol{\vartheta} - \tilde{\boldsymbol{F}}(\omega)] \leqslant 0, \tag{2.33}$$

with

$$\begin{aligned} &\boldsymbol{n} = \tilde{\boldsymbol{F}}_2^* - \tilde{\boldsymbol{F}}_1^* = [1, -1]^T, \\ &\boldsymbol{\vartheta} = [\tilde{F}_1, \;\; \tilde{F}_2]^T, \\ &\tilde{\boldsymbol{F}}(\omega) = (1-\omega)\, \tilde{\boldsymbol{F}}_1^* + \omega\, \tilde{\boldsymbol{F}}_2^*, \;\; 0 \leqslant \omega \leqslant 1, \end{aligned} \tag{2.34}$$

where, ω is the weighting factor adjusted to give the solutions with a specified trade-off.

2.3.2 Divide and conquer algorithm

The Pareto front must include a limited number of non-dominated solutions, which possess a significant trade-off between conflicting objectives. Decision-makers are more interested in

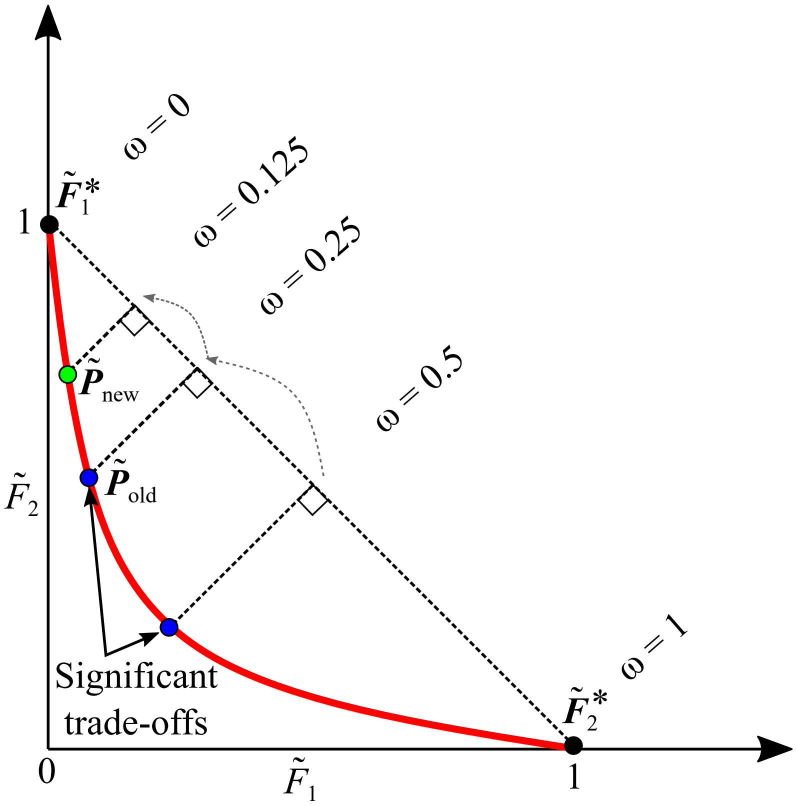


Fig. 2.6: Demonstration of the D&C algorithm for the bi-objective case.

steep segments of the Pareto front, i.e., knees, due to the high trade-off level compared to the flat segments. Filtering methods were used to keep the solutions with significant trade-offs from the dense Pareto fronts [76, 77]. However, these methods suffer from producing insignificant solutions ahead; therefore, they are computationally expensive. Hashem *et al.* [78] proposed the D&C algorithm that breaks down the problem into two sub-problems until each one satisfies a single solution with a specified trade-off. This algorithm can be combined together with one of the optimization methods, (e.g., NNC [75] or Normal Boundary Intersection (NBI) [79]) to explore the significant Pareto optimal solutions with an adaptive manner.

The working principle for the two-dimensional case is illustrated in Fig. 2.6. First, The algorithm divides the Pareto front into two segments by setting a weight that divides the Utopia line in half. After that, the Pareto optimal solution is calculated. The algorithm continues subdividing the previous segments and calculating the mid-point solutions until the new solution becomes insignificant compared to the previous solution. The user determines the significance criterion of the algorithm. For instance, the new point $\boldsymbol{P}_{\text{new}}(P_{\text{new},1}, P_{\text{new},2})$ is considered significant relative to previous point $\boldsymbol{P}_{\text{old}}(P_{\text{old},1}, P_{\text{old},2})$ if the following constraint is satisfied:

$$\min\{(|P_{\text{new},1} - P_{\text{old},1}|, |P_{\text{new},2} - P_{\text{old},2}|)\} \geqslant P_{\text{limit}}, \tag{2.35}$$

where P_{limit} is the user-defined trade-off tolerance.

Chapter 3

Kinematically Constrained Reparameterization for Optimal Time and Jerk Motion

This chapter presents a kinematically constrained spline-based reparameterization approach for the trajectory generation of industrial machines while maintaining the geometric shape. This method is targeted for optimal industrial product designs which require an exact representation of the geometric path, e.g., CAD design. It is a decoupled approach to parameterizing the trajectory in terms of splines and formulates an OCP as a BOOP. The properties of splines are exploited to ensure kinematic constraint satisfaction for all times. Two objectives are considered herein: total time and jerk square integral of the trajectory. The Pareto front comprising the trade-off solutions, which represent the contradictory nature of both objectives, is explored. The results are compared with the widely used linear reparameterization method by simulations and experiments.

In this chapter, a parametric curve with jerk continuity is used to represent the given via-points in the Cartesian space; therefore, the predefined geometric path is obtained. As in [80], the reparameterization function in terms of the B-spline is adopted to have a nonlinear relationship between the curve parameter and motion time. Considering the total time as an unknown parameter, the computation for the jerk square integral of the reparameterized trajectory is very expensive; hence the trapezoidal method is used to estimate the jerk value at each partitioning interval. A finite set of inequality constraints are determined based on the convex hull of the reparameterization function derivatives and the maximum parametric curve derivatives to satisfy the kinematic limits along the trajectory. Moreover, adding equality constraints at the start and end of the reparameterization function allows smooth start- and end-transitions (i.e., zero initial and final velocities and accelerations). For the BOOP, the process of revealing the Pareto front comprising trade-off solutions between time and jerk is implemented by applying the D&C [78, 81, 82] with the NNC method [75, 83], where each solution is computed using the SQP [84]. This achieves an efficient representation of the Pareto front and determines the constrained reparameterization of the trajectory with a significant trade-off.

The rest of this chapter is organized as follows: Section 3.1 describes the related works with the proposed method; Section 3.2 briefly describes the representation of the given via-points as a predefined geometric path by interpolation or approximation algorithms and trajectory reparameterization by a B-spline; Section 3.3 proposes the method for limiting the kinematic constraints for the reparameterized trajectory; Section 3.4 states the bi-objective optimization scheme for time and jerk, followed by the application of the NNC method and the D&C algorithm; Section 3.5 presents the effectiveness of the proposed method as validated by simulation and experimental results; and Section 3.6 draws the summary of this chapter.

3.1 Related works

In Section 1.2.3, several OCPs concerning with motion time and jerk of the trajectory are discussed. Also, techniques to solve the OCPs are addressed. Most importantly, to simultaneously satisfy geometric positions of given via-points and desired smoothness of a contour profile, a smooth interpolation or approximation function with sufficient continuity is required to parameterize the motion trajectory of industrial machines.

Splines are piecewise polynomial functions widely used to generate smooth trajectory primitives to ensure acceleration or jerk continuity [85]. The spline representation in terms of a compact basis form known as B-spline gives a more simple computation and a precise representation of smooth free-form curves and surfaces [25]. The interpolation or approximation of via-points is independent of the B-spline order. For general usage of splines, the required geometric path is usually generated in a parametric form, and the specific reparameterization of the curve is applied to modify the motion trajectories [14, 28, 86, 87]. However, these methods initially require computing the arc length of the parametric curve to determine the required time and velocity profile. The curve parameter is then updated in each time step. As a drawback, the excitation of the machine vibration may occur due to the inaccurate mapping between the curve parameter and the arc displacement.

The geometric path of a robot usually consists of a set of points, called via-points, in the Cartesian space, where the tasks to perform and obstacles to avoid are determined. In this case, trajectory planning takes an input of the geometric path, kinematic, and dynamic limitations to industrial machines and generates an output as a reference trajectory either in a joint or Cartesian space. Therefore, the optimal trajectory generation of a geometric path is generally translated into an OCP while considering the equality and inequality constraints of industrial machines. Spline-based trajectory generation problem is formulated as an OCP using the spline properties for constraint satisfaction.

Regarding the spline-based OCPs, mostly, the optimal trajectories were generated based on changing the geometric shape to achieve the desired smoothness or time criteria. One exception is the work of Hashemian *et al.* [80], which presented a method for minimizing the total jerk of the trajectory while maintaining the geometric shape at a predefined time. In this method, the geometric path was reparameterized in time by a B-spline, and the nonlinear relationship between the curve parameter and time was formulated by jerk minimization. However, time must be considered as a priori, and the kinematic constraints along the trajectory were not addressed.

3.2 Geometric path representation and trajectory reparameterization

B-splines are widely used for the geometric modeling of curves and surfaces due to their flexibility. The construction and properties of the B-splines are discussed in detail [11]. Given via-points, the following two types of curve fitting methods are considered to represent the geometric path: interpolation, where the curve passes through all via-points, and approximation, where the curve passes near the via-points by minimizing the error between them. The parametric B-spline of order k consisting of piecewise polynomial functions of degree $k-1$ and $n+1$ control points is determined as follows:

$$\boldsymbol{s}(u) = \sum_{j=0}^{n} B_{j,k}(u)\ \boldsymbol{p}_j, \ \ 0 \leqslant u \leqslant 1, \tag{3.1}$$

with a non-decreasing knot vector

$$U = [\ \underbrace{0,\ ...,\ 0}_{k\text{-times}},\ u_k, u_{k+1}, ..., u_n,\ \underbrace{1,\ ...,\ 1}_{k\text{-times}}\ ], \tag{3.2}$$

where, $\boldsymbol{s}(u)$ represents the position of the geometric path with parameter u; $\boldsymbol{p}_j$ denotes the three-dimensional position control point; and the knot vector U consists of $n+k+1$ knots that are k-times clamped at both ends. The first and last position control points coincide with the first and last via-points, respectively. The distribution of the inner knots in (3.2) reflects the interpolation or approximation curve to be fitted [11]. The global interpolation and approximation algorithms refer to Section 2.2. The unknown position coefficients for the parametric curve can be solved in (3.1) by knowing the basis functions and the via-points to be interpolated or approximated. Thereafter, the parametric curve derivatives can be calculated according to (2.9). In this study, the sixth-order parametric B-splines are used to represent the geometric paths to generate trajectories with a continuous jerk.

After the geometric path is defined as a parametric curve $\boldsymbol{s}$(u), a particular reparameterization method is generally adopted for obtaining motion trajectories with respect to time. Reparameterization refers to velocity, acceleration, and jerk vectors modification without changing the geometric shape. More specifically, the curve parameter u is expressed in terms of time t by a strictly monotonic increasing function $u(t)$. Therefore, the curves $\boldsymbol{s}(u)$ and $\tilde{\boldsymbol{s}}(u(t))$ are geometrically the same in position, but are kinematically different functions. Reparameterization of the geometric path is demonstrated in Fig. 3.1. The velocity, acceleration, and jerk of the reparameterized trajectory can be expressed in terms of time by the chain rule as follows:

$$\tilde{\boldsymbol{v}}(t) = \dot{\boldsymbol{s}}(u)\, \dot{u}(t), \tag{3.3}$$

$$\tilde{\boldsymbol{a}}(t) = \ddot{\boldsymbol{s}}(u)\,[\dot{u}(t)]^2 + \dot{\boldsymbol{s}}(u)\,\ddot{u}(t), \tag{3.4}$$

$$\tilde{\boldsymbol{j}}(t) = \dddot{\boldsymbol{s}}(u)\,[\dot{u}(t)]^3 + 3\,\ddot{\boldsymbol{s}}(u)\,\ddot{u}(t)\,\dot{u}(t) + \dot{\boldsymbol{s}}(u)\,\dddot{u}(t), \tag{3.5}$$

where $\tilde{\boldsymbol{v}}(t)$, $\tilde{\boldsymbol{a}}(t)$, and $\tilde{\boldsymbol{j}}(t)$ are the respective velocity, acceleration, and jerk of the reparameterized trajectory, $\dot{\boldsymbol{s}}(u)$, $\ddot{\boldsymbol{s}}(u)$, and $\dddot{\boldsymbol{s}}(u)$ are the first, second, and third derivatives of the geometric path with respect to parameter u, and $\dot{u}(t)$, $\ddot{u}(t)$, and $\dddot{u}(t)$ are the pseudo velocity, acceleration, and jerk with respective to time t. The reparameterized trajectory is required to satisfy the upper and lower bounds on kinematic limits along the trajectory as follows:

$$\boldsymbol{v}_{\min} \leqslant \tilde{\boldsymbol{v}}(t) \leqslant \boldsymbol{v}_{\max} \tag{3.6}$$

$$\boldsymbol{a}_{\min} \leqslant \tilde{\boldsymbol{a}}(t) \leqslant \boldsymbol{a}_{\max} \tag{3.7}$$

$$\boldsymbol{j}_{\min} \leqslant \tilde{\boldsymbol{j}}(t) \leqslant \boldsymbol{j}_{\max} \tag{3.8}$$

where $\boldsymbol{v}_{\min}$, $\boldsymbol{a}_{\min}$, and $\boldsymbol{j}_{\min}$, and $\boldsymbol{v}_{\max}$, $\boldsymbol{a}_{\max}$, and $\boldsymbol{j}_{\max}$ are lower and upper bounds on velocity, acceleration, and jerk of the trajectory, respectively.

3.2.1 Representation of reparameterization function by B-spline

The reparameterization function for the parametric curve is adopted herein in terms of a piecewise continuous function by a B-spline [80]. The q^{th} order B-spline reparameterizaion function with $m+1$ control points is given as follows:

$$u(t) = \sum_{i=0}^{m} B_{i,q}(t)\;\; c_i^{\text{pos}},\;\; 0 \leqslant t \leqslant t_{\text{f}}, \tag{3.9}$$

with its knot vector

$$T = [\;\underbrace{0,\;...,\;0}_{q\text{-times}},\; t_q, t_{q+1}, ..., t_m,\; \underbrace{t_{\text{f}},\;...,\;t_{\text{f}}}_{q\text{-times}}\;], \tag{3.10}$$

where c_i^{pos} represents the scalar position coefficient of the reparameterization function, and t_{f} denotes the unknown total time. The knot vector T is clamped q-times at 0 and t_{f}, hence, the first and last position coefficients of the reparameterization function coincide with the first and last curve parameter values, $c_0^{\text{pos}} = 0$ and $c_m^{\text{pos}} = 1$, respectively. The other position coefficients

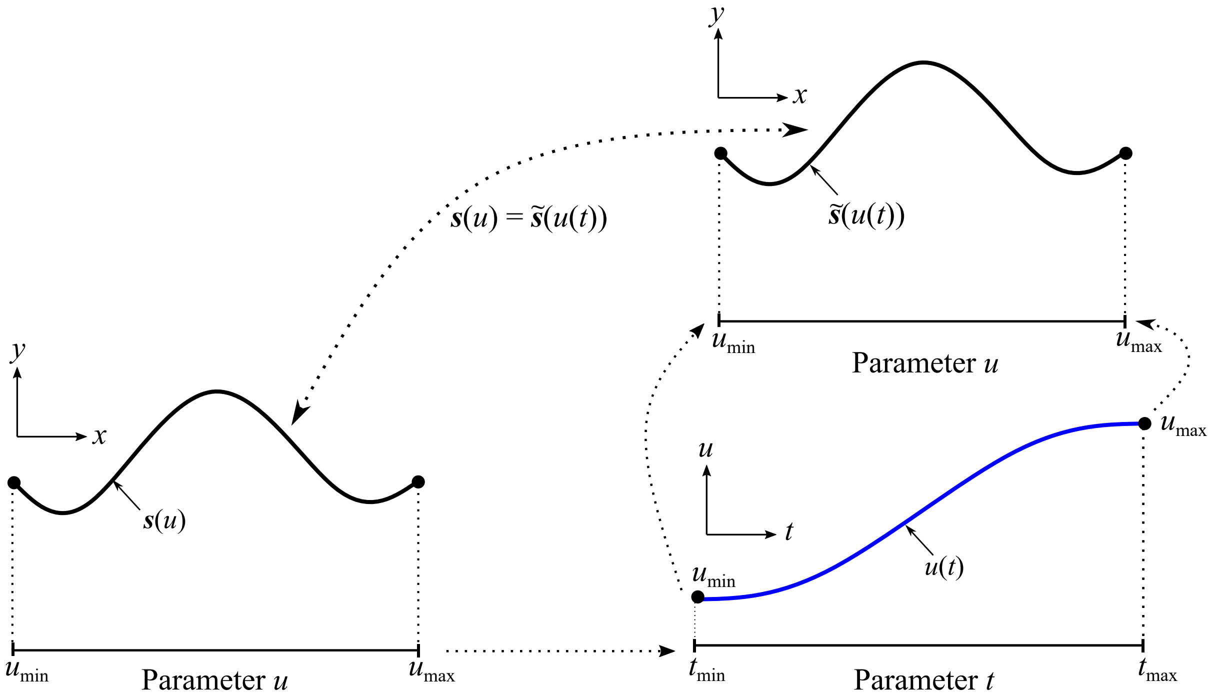


Fig. 3.1: Reparameterization of the geometric path.

lie between 0 and 1, and the inner knots in (3.10) are assumed to be uniformly distributed between the interval $[0, t_{\mathrm{f}}]$. Therefore, the unknown total time and position coefficients of the reparameterization function are determined through the bi-objective optimization procedure presented in Section 3.4.

3.2.2 Convex hull property by B-spline

The convex hull property of B-spline states that the piecewise segment of the curve lies within the convex hull of its coefficients. Therefore, the segment of the reparameterization function $u(t)$, $t \in [t_i, t_{i+1}]$ must exist within the convex hull formed by the position coefficients $c^{\mathrm{pos}}_{i-q+1}, ..., c^{\mathrm{pos}}_{i}$ and the change in these points locally affects the function shape. The reparameterization function $u(t)$ is a B-spline; thus, the pseudo velocity, acceleration, and jerk are also B-splines, and they must be determined separately to define the kinematic constraints for the trajectory. The pseudo velocity is reconstructed as follows by discarding the first and last values of the knot vector in (3.10):

$$\dot{u}(t) = \sum_{i=0}^{m-1} B_{i,q-1}(t)\ c_i^{\text{vel}}, \tag{3.11}$$

with

$$c_i^{\text{vel}} = \frac{(q-1)}{t_{i+q} - t_{i+1}}(c_{i+1}^{\text{pos}} - c_i^{\text{pos}}),\ \ i = \{0, 1, ..., m-1\}. \tag{3.12}$$

Similarly, the acceleration of the reparameterization function is given by

$$\ddot{u}(t) = \sum_{i=0}^{m-2} B_{i,q-2}(t)\ c_i^{\text{acc}}, \tag{3.13}$$

with

$$c_i^{\text{acc}} = \frac{(q-2)}{t_{i+q} - t_{i+2}}(c_{i+1}^{\text{vel}} - c_i^{\text{vel}}),\ \ i = \{0, 1, ..., m-2\}. \tag{3.14}$$

The jerk function is defined as follows:

$$\dddot{u}(t) = \sum_{i=0}^{m-3} B_{i,q-3}(t)\ c_i^{\text{jerk}}, \tag{3.15}$$

with

$$c_i^{\text{jerk}} = \frac{(q-3)}{t_{i+q} - t_{i+3}}(c_{i+1}^{\text{acc}} - c_i^{\text{acc}}),\ \ i = \{0, 1, ..., m-3\}. \tag{3.16}$$

where, c_i^{vel}, c_i^{acc}, and c_i^{jerk} represent the control points of pseudo velocity, acceleration, and jerk, respectively. Fig. 3.2 depicts the formation of those pseudo-velocity, -acceleration, and -jerk control points, which are the convex hull vertices of (3.11), (3.13), and (3.15), respectively.

3.3 Constrained kinematic reparameterization

The initial and final velocity and acceleration of the trajectory must be zero to obtain a smooth transition at the start and end of the trajectory. Hence, the boundary conditions for velocity

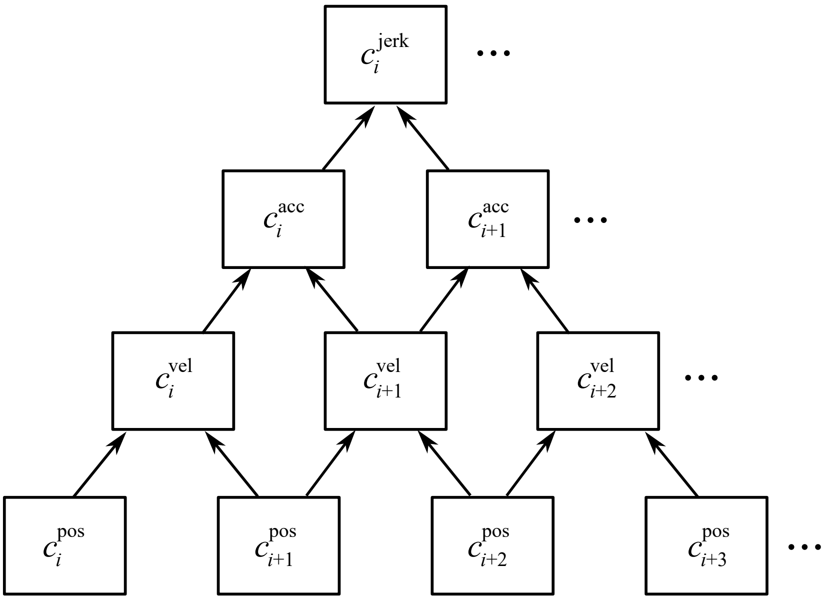


Fig. 3.2: Determination of pseudo-velocity, -acceleration, and -jerk control points in terms of position control points of the reparameterization function to define a finite set of constraints.

and acceleration are added as equality constraints of the reparameterized trajectory in (3.3) and (3.4) as follows:

$$\begin{aligned}
\tilde{\boldsymbol{v}}(0) &= \dot{\boldsymbol{s}}(0)\ c_0^{\text{vel}} = 0. \\
\tilde{\boldsymbol{v}}(t_{\text{f}}) &= \dot{\boldsymbol{s}}(1)\ c_{m-1}^{\text{vel}} = 0. \\
\tilde{\boldsymbol{a}}(0) &= \ddot{\boldsymbol{s}}(0)\ \left[c_0^{\text{vel}}\right]^2 + \dot{\boldsymbol{s}}(0)\, c_0^{\text{acc}} = 0. \\
\tilde{\boldsymbol{a}}(t_{\text{f}}) &= \ddot{\boldsymbol{s}}(1)\ \left[c_{m-1}^{\text{vel}}\right]^2 + \dot{\boldsymbol{s}}(1)\, c_{m-2}^{\text{acc}} = 0.
\end{aligned} \tag{3.17}$$

Moreover, the inequality constraints of the velocity, acceleration, and jerk must be determined such that the reparameterized trajectory satisfies the kinematic limits. Here, the change in the pseudo-velocity, -acceleration, and -jerk affects the kinematic values of the trajectory. Therefore, the inequality constraints are proposed based on the pseudo-velocity, -acceleration, and -jerk control points and the maximum derivatives of the parametric curve. If the maximum and minimum kinematic limits are assumed to be symmetric for all axes, a finite set of velocity constraints are proposed to satisfy (3.6) as follows:

$$\tilde{\boldsymbol{v}}_i \leqslant \boldsymbol{v}_{\text{lim}},\ i = \{0, 1, ..., m-1\}, \tag{3.18}$$

with

$$\tilde{\boldsymbol{v}}_i = |\dot{\boldsymbol{s}}(u)|_{\text{max}}\, |c_i^{\text{vel}}|, \tag{3.19}$$

where $|\dot{\boldsymbol{s}}(u)|_{\text{max}}$ is the maximum absolute first derivative of parametric curve and $\boldsymbol{v}_{\text{lim}}$ denotes the absolute velocity limit for the trajectory.

Pseudo-velocity and -acceleration control points are required to define the acceleration constraints. Since the pseudo-acceleration control point c_i^{acc} is the rate of change of two velocity control points, c_i^{vel} and c_{i+1}^{vel}, there are two combinations for each i^{th} control point of pseudo-acceleration, which are required to satisfy (3.7) as follows:

$$\tilde{\boldsymbol{a}}_{i,l} \leqslant \boldsymbol{a}_{\text{lim}},\ i = \{0, 1, ..., m-2\},\ l = \{0, 1\}, \tag{3.20}$$

with

$$\tilde{\boldsymbol{a}}_{i,l} = |\ddot{\boldsymbol{s}}(u)|_{\text{max}}\, |c_{i+l}^{\text{vel}}|^2 + |\dot{\boldsymbol{s}}(u)|_{\text{max}}\, |c_i^{\text{acc}}|, \tag{3.21}$$

where $|\ddot{\boldsymbol{s}}(u)|_{\text{max}}$ is the maximum absolute second derivative of parametric curve, and $\boldsymbol{a}_{\text{lim}}$ denotes the absolute acceleration limit. Similarly, the pseudo-jerk control point c_i^{jerk} is the rate of change of c_i^{acc} and c_{i+1}^{acc}, which again consist of three velocity control points, namely c_i^{vel}, c_{i+1}^{vel}, and c_{i+2}^{vel}, respectively. Therefore, four combinations for each i^{th} control point of pseudo-jerk are required to satisfy (3.8) as follows:

$$\tilde{\boldsymbol{j}}_{i,l} \leqslant \boldsymbol{j}_{\text{lim}},\ i = \{0, 1, ..., m-3\},\ l = \{0, 1, 2, 3\}, \tag{3.22}$$

with

$$\tilde{\boldsymbol{j}}_{i,l} = \begin{cases} |\dddot{\boldsymbol{s}}(u)|_{\max} \, |c_{i+l}^{\text{vel}}|^3 + 3\, |\ddot{\boldsymbol{s}}(u)|_{\max} \, |c_i^{\text{acc}}| \, |c_{i+l}^{\text{vel}}| \\ + |\dot{\boldsymbol{s}}(u)|_{\max} \, |c_i^{\text{jerk}}|, \text{ for } l \leqslant 1, \\ |\dddot{\boldsymbol{s}}(u)|_{\max} \, |c_{i+l-1}^{\text{vel}}|^3 + 3\, |\ddot{\boldsymbol{s}}(u)|_{\max} \, |c_{i+1}^{\text{acc}}| \, |c_{i+l-1}^{\text{vel}}| \\ + |\dot{\boldsymbol{s}}(u)|_{\max} \, |c_i^{\text{jerk}}|, \text{ otherwise,} \end{cases} \tag{3.23}$$

where $|\dddot{\boldsymbol{s}}(u)|_{\max}$ and $\boldsymbol{j}_{\text{lim}}$ represent the maximum absolute third derivative of the parametric curve and the absolute jerk limit of the trajectory, respectively.

3.4 Bi-objective optimization approach

3.4.1 Objective functions and constraints

This section formulates an OCP as the bi-objective optimization problem for time and jerk of the reparameterized trajectory. The objective functions are adopted with two contradictory terms: the total time and the jerk square integral of the reparameterized trajectory. Moreover, the kinematic constraints of the trajectory described in Section 3.3 are considered as equality and inequality constraints for the optimization. Therefore, the problem is formulated as follows:

$$\min_{\phi} \{F_1, F_2\}, \tag{3.24}$$

where

$$\begin{aligned} F_1 &= t_{\text{f}}, \\ F_2 &= \int_0^{t_{\text{f}}} \left\| \tilde{\boldsymbol{j}}(t) \right\|_2^2 dt, \\ \phi &= \left[t_{\text{f}}, c_0^{\text{pos}}, c_1^{\text{pos}}, ..., c_m^{\text{pos}}\right], \end{aligned} \tag{3.25}$$

subject to

$$
\begin{aligned}
& c_0^{\text{pos}} = 0, \;\; c_m^{\text{pos}} = 1, \\
& c_0^{\text{pos}} \leqslant c_1^{\text{pos}} \leqslant \dots \leqslant c_{m-1}^{\text{pos}} \leqslant c_m^{\text{pos}}, \\
& 0 < t_{\text{f}} \leqslant t_{\max}, \\
& \tilde{\boldsymbol{v}}(0) = 0, \;\; \tilde{\boldsymbol{v}}(t_{\text{f}}) = 0, \\
& \tilde{\boldsymbol{a}}(0) = 0, \;\; \tilde{\boldsymbol{a}}(t_{\text{f}}) = 0, \\
& \tilde{\boldsymbol{v}}_i \leqslant \boldsymbol{v}_{\text{lim}}, \;\; i = \{0, 1, \dots, m-1\}, \\
& \tilde{\boldsymbol{a}}_{i,l} \leqslant \boldsymbol{a}_{\text{lim}}, \; i = \{0, 1, \dots, m-2\}, \; l = \{0, 1\}, \\
& \tilde{\boldsymbol{j}}_{i,l} \leqslant \boldsymbol{j}_{\text{lim}}, \;\; i = \{0, 1, \dots, m-3\}, \; l = \{0, 1, 2, 3\},
\end{aligned}
\tag{3.26}
$$

where $\|.\|_2$ is the Euclidean norm, and $t_{\max}$ is the maximum time limit defined by the user. Jerk is the third derivative of position; thus, the orders for the parametric curve and the reparameterization function must be chosen as sixth or higher to have a continuous jerk profile. In other words, the F_2 computation is very expensive. Here, the jerk square integral of the reparameterized trajectory is estimated by the trapezoidal method as follows:

$$
\int_0^{t_{\text{f}}} \left\| \tilde{\boldsymbol{j}}(t) \right\|_2^2 dt \approx \sum_{h=1}^{N} \frac{\left\| \tilde{\boldsymbol{j}}(t_{h-1}) \right\|_2^2 + \left\| \tilde{\boldsymbol{j}}(t_h) \right\|_2^2}{2N} t_{\text{f}}, \tag{3.27}
$$

where $\tilde{\boldsymbol{j}}(t_h)$ is the jerk value at each time step t_h, and N is the number of partitions of the function.

3.4.2 Generation of Pareto optimal solutions

For the BOOP, optimal solutions are presented as the trade-off solutions between total time and jerk square integral by the Pareto front. Pareto optimality conditions can be found in [74, 88]. The extrema of the objective set in (3.24) are formulated by minimizing each objective independently as follows:

$$
f_{1,\min} = \min_{\phi} F_1, \quad f_{2,\min} = \min_{\phi} F_2, \tag{3.28}
$$

subject to (3.26).

The solutions in (3.28) are mapped into the normalized objective space formulated by

$$\tilde{F}_1 = \frac{F_1 - f_{1,\min}}{f_{1,\max} - f_{1,\min}}, \;\; \tilde{F}_2 = \frac{F_2 - f_{2,\min}}{f_{2,\max} - f_{2,\min}}, \tag{3.29}$$

with

$$\begin{aligned} f_{1,\max} = t_{\max}, \qquad & f_{2,\max} = F_2(\phi_{1,\min}), \\ & \phi_{1,\min} = \arg\min_{\phi} F_1. \end{aligned} \tag{3.30}$$

The anchor points $\tilde{\boldsymbol{F}}_1^*$ and $\tilde{\boldsymbol{F}}_2^*$, which are the extreme coordinates in a normalized plane, are represented as $(0,1)$ and $(1,0)$, respectively by solving (3.29) and (3.30). Therefore, the optimization problem in (3.24) is reformulated as follows by the NNC method:

$$\min_{\phi} \tilde{F}_2, \tag{3.31}$$

subject to (3.26), and:

$$\boldsymbol{n}^T[\boldsymbol{\vartheta} - \tilde{\boldsymbol{F}}(\omega)] \leqslant 0, \tag{3.32}$$

with

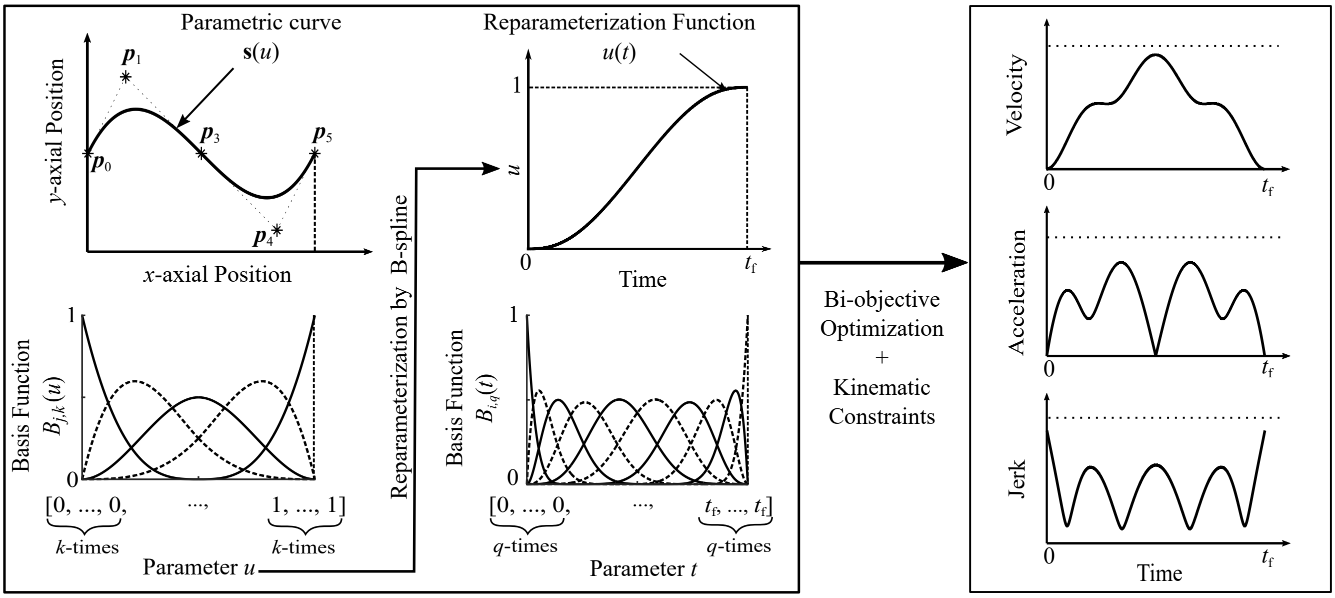
$$\begin{aligned} &\boldsymbol{n} = \tilde{\boldsymbol{F}}_2^* - \tilde{\boldsymbol{F}}_1^* = [1, -1]^T, \\ &\boldsymbol{\vartheta} = [\tilde{F}_1, \;\; \tilde{F}_2]^T, \\ &\tilde{\boldsymbol{F}}(\omega) = (1-\omega)\,\tilde{\boldsymbol{F}}_1^* + \omega\,\tilde{\boldsymbol{F}}_2^*, \;\; 0 \leqslant \omega \leqslant 1, \end{aligned} \tag{3.33}$$

where ω is the weighting factor adjusted to give the solutions with a specified trade-off.

Thereafter, the D&C algorithm is used to find a set of Pareto-relevant solutions $\boldsymbol{\Theta}$ with a minimum significant trade-off tolerance between consecutive trade-off solutions. The best trade-off solution $\boldsymbol{\vartheta}^*$ is chosen by the following condition

$$\boldsymbol{\vartheta}^* = \arg\min_{\boldsymbol{\vartheta}} ||\boldsymbol{\Theta}||_2, \tag{3.34}$$

corresponding to the weighting factor ω^*.



Geometric Path Representation (Parametric Curve) and Reparameterization by B-spline

Optimized Trajectory

Fig. 3.3: Overview of the B-spline reparameterization from the curve parameter u to motion time and bi-objective optimization with kinematic constraints.

Fig. 3.3 depicts an overview of proposed B-spline reparameterization and bi-objective optimization method.

3.5 Execution of algorithm

3.5.1 Calculation conditions

In this section, the proposed kinematically constrained reparameterization with a bi-objective optimization for time and jerk is investigated with two geometric paths: S-shaped and GEMINI airfoil profiles. Thereafter, the benefit of the B-spline reparameterization over the linear reparameterization is discussed with an S-shaped profile. The optimization problems in Section 3.4 are solved using the SQP ("fmincon" function) in a MATLAB® environment of Core i7-7500U Central Processing Unit (CPU) and 8 GB Random Access Memory (RAM) laptop computer with a Windows 10 64-bit operating system.

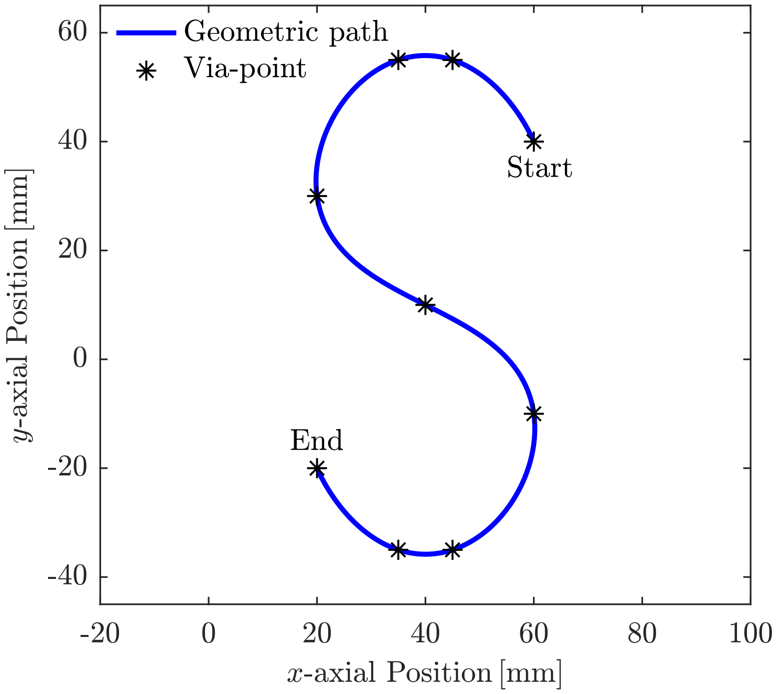


Fig. 3.4: Geometric path representation of the S-shaped profile by a sixth-order parametric B-spline interpolating nine via-points.

3.5.2 S-shaped profile

Given nine via-points in an x-y plane, the geometric path is interpolated by the sixth-order parametric B-splines with nine control points. Fig. 3.4 illustrates the positions of the predefined geometric path as a parametric curve of the S-shaped profile. For this example, the velocity, acceleration, and jerk limits of optimization are set as $\boldsymbol{v}_{\text{lim}} = [80, 80]$ mm/s, $\boldsymbol{a}_{\text{lim}} = [500, 500]$ mm/s^2, and $\boldsymbol{j}_{\text{lim}} = [20000, 20000]$ mm/s^3, respectively. The maximum time limit for the trajectory is defined as $t_{\text{max}} = 6$ s.

The reparameterization function is chosen as the sixth-order B-spline function with 16 control points. The number of partitions $N = 1000$ is used to estimate the jerk square integral. The minimum trade-off tolerance for the D&C algorithm is set as 0.01. Fig. 3.5 represents the Pareto front consisting of significant solutions between total time and jerk square integral. The solution for the optimal time is achieved with the highest possible jerk value that satisfies the kinematic limits of the trajectory. Similarly, the solution with the lowest jerk value is found at $t_{\text{f}} = t_{\text{max}}$. The best trade-off solution occurs at time $t_{\text{f}} = 3.63$ s with the weighting

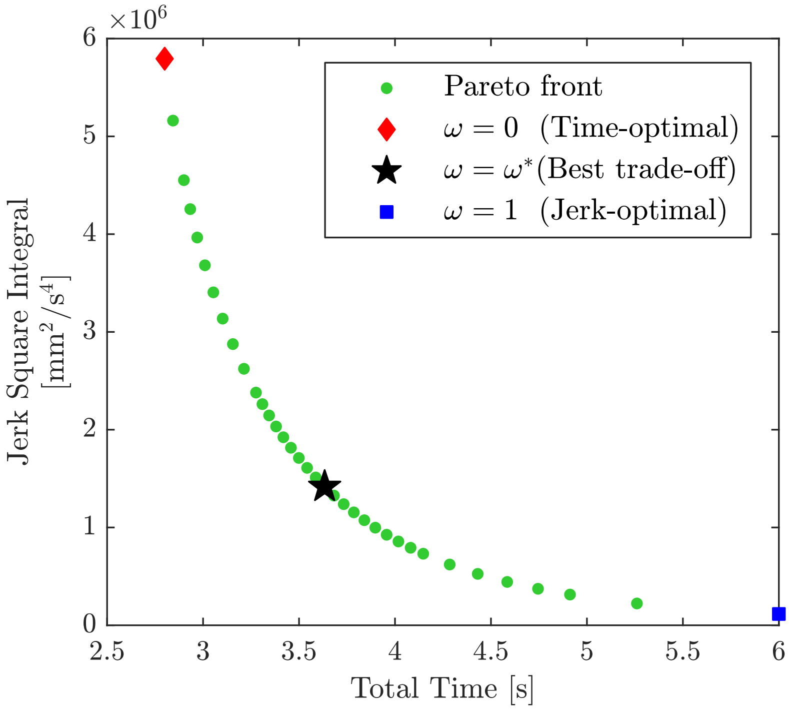


Fig. 3.5: Pareto front representing the optimization results of the S-shaped profile.

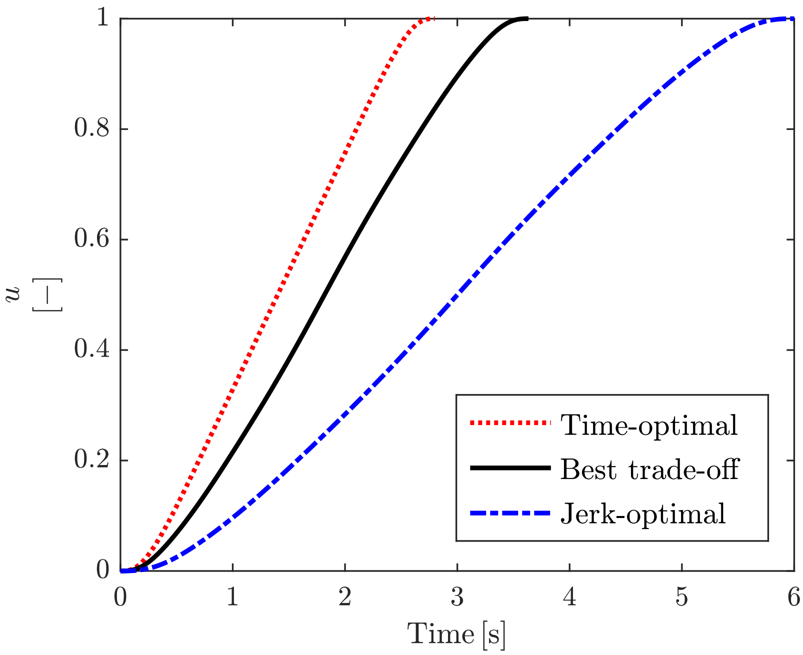


Fig. 3.6: Relation between parameter u and motion time in terms of B-spline reparameterization function for time-optimal, best trade-off, and jerk-optimal results of the S-shaped profile.

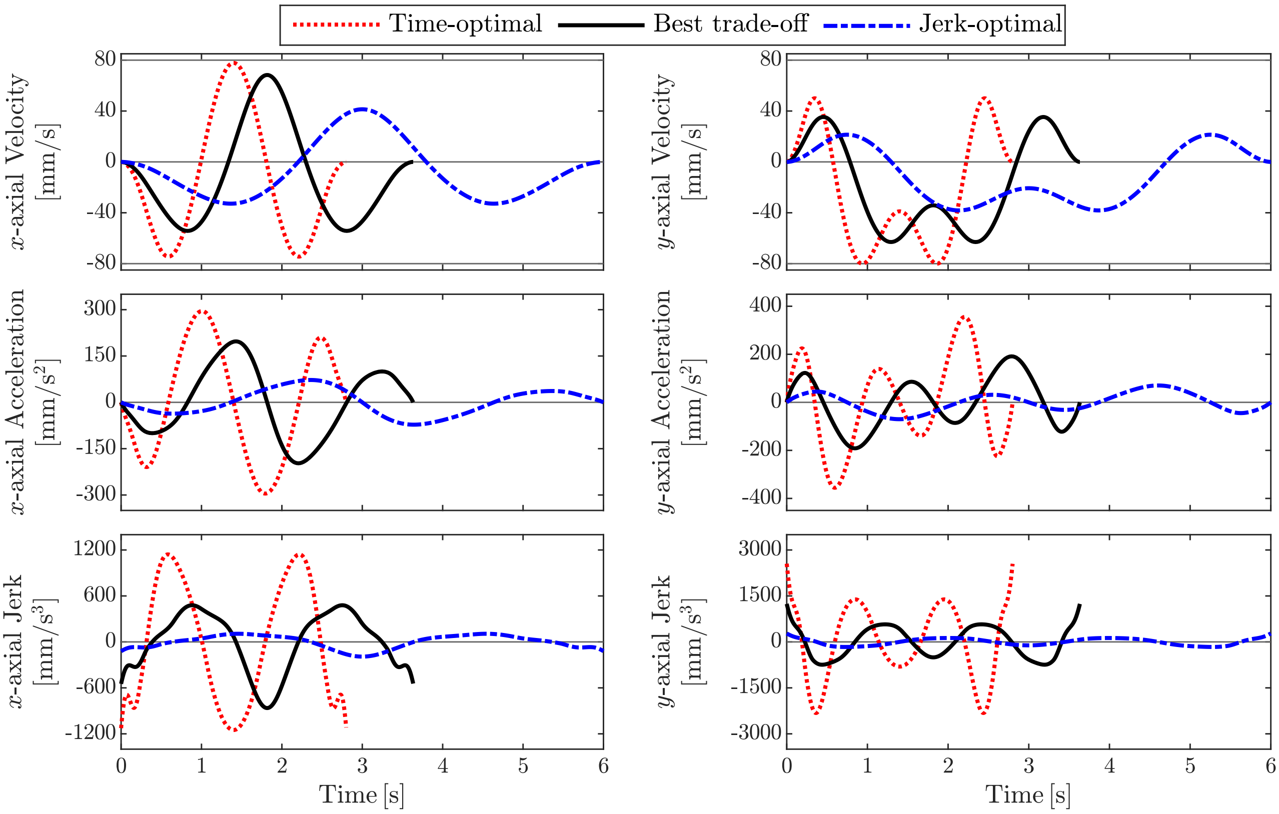


Fig. 3.7: Satisfaction of velocity, acceleration, and jerk limits for x- and y-axis of the S-shaped profile.

factor $\omega^* = 0.51$, and approximately 75% of the highest jerk value can be reduced by using approximately 60% of $t_{\max}$.

Three conditions of the relation between parameter u and motion time are provided to investigate the effect of the B-spline reparameterization on bi-objective optimization; time-optimal, jerk-optimal, and best trade-off shown in Fig. 3.6. The generated motion can be faster or slower depending on the reparameterization function to optimize the trajectory. The parameter u is transversed slowly at the start and end of the motion for all cases to avoid abrupt changes in velocity and acceleration values, which achieves smooth start- and end-transitions. For the rest of the motion in the time-optimal case, the reparameterization function finds the fastest way for traversing parameter u from 0 to 1. For the jerk-optimal case, the function gives more flexible reparameterization of the trajectory by modifying its position coefficients to provide the lowest jerk value at $t_{\max}$. The reparameterization function finds an optimal solution, which considers time and jerk equally as the best trade-off case.

Fig. 3.7 presents a comparison of the velocity, acceleration, and jerk in x- and y-axis for time-optimal, best trade-off, and jerk-optimal trajectories. With the proposed method, the initial and final velocity and acceleration of the trajectory become zero. Moreover, the velocity, acceleration, and jerk limits are satisfied in each axis along the trajectory of the S-shaped profile.

3.5.3 GEMINI profile

For the second example, the GEMINI profile with a chord length of 100 mm, which consists of 79 points, are downloaded from the UIUC airfoil database [89]. These points are adopted as the via-points. The sixth-order parametric B-spline with 12 control points is used to approximate the geometric path (Fig. 3.8). In this case, the curve does not exactly pass through all the via-points, but the geometric path is approximated in the sense of least squares with the total error of $[123.34, 8.17]$ mm^2. For this trajectory, the axial velocity, acceleration, and jerk limits are determined as $\boldsymbol{v}_{\text{lim}} = [150, 150]$ mm/s, $\boldsymbol{a}_{\text{lim}} = [1000, 1000]$ mm/s^2, and $\boldsymbol{j}_{\text{lim}} = [30000, 30000]$ mm/s^3, respectively. The maximum time limit is set as $t_{\text{max}} = 8$ s.

The reparameterization function, number of partitions, and minimum trade-off tolerance for the D&C algorithm are same to those in the previous example (Section 3.5.2). Fig. 3.9 shows the Pareto front representation between total time and jerk square integral of the GEMINI profile consisting of significant trade-off solutions. The best trade-off solution is found at $t_{\text{f}} = 4.52\,\text{s}$ with the weighting factor of $\omega^* = 0.52$ and has approximately 43% of the time saving potential compared to the jerk-optimal case and approximately 81% of the jerk saving potential compared to the time-optimal case. Fig. 3.10 illustrates the satisfaction of the velocity, acceleration, and jerk limits of the GEMINI airfoil profile in x- and y-axis, respectively. Zero start- and end-velocities and accelerations are also satisfied by the proposed method.

3.5.4 Comparison

The simulation results were compared with an S-shaped profile to investigate the effect of the B-spline reparameterization over the linear reparameterization on the bi-objective optimization.

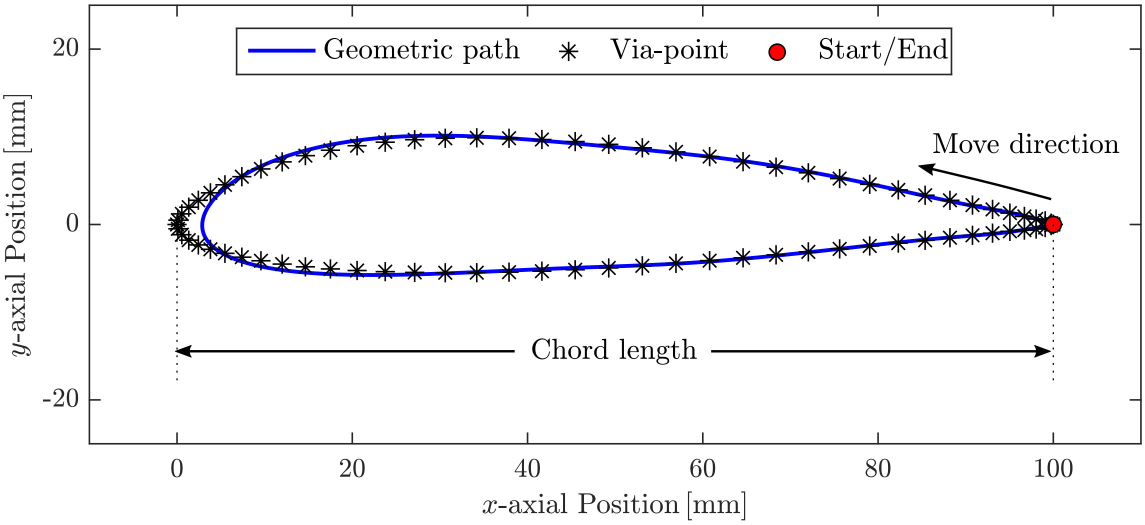


Fig. 3.8: Geometric path representation of the GEMINI airfoil profile by a sixth-order parametric B-spline approximating 79 via-points.

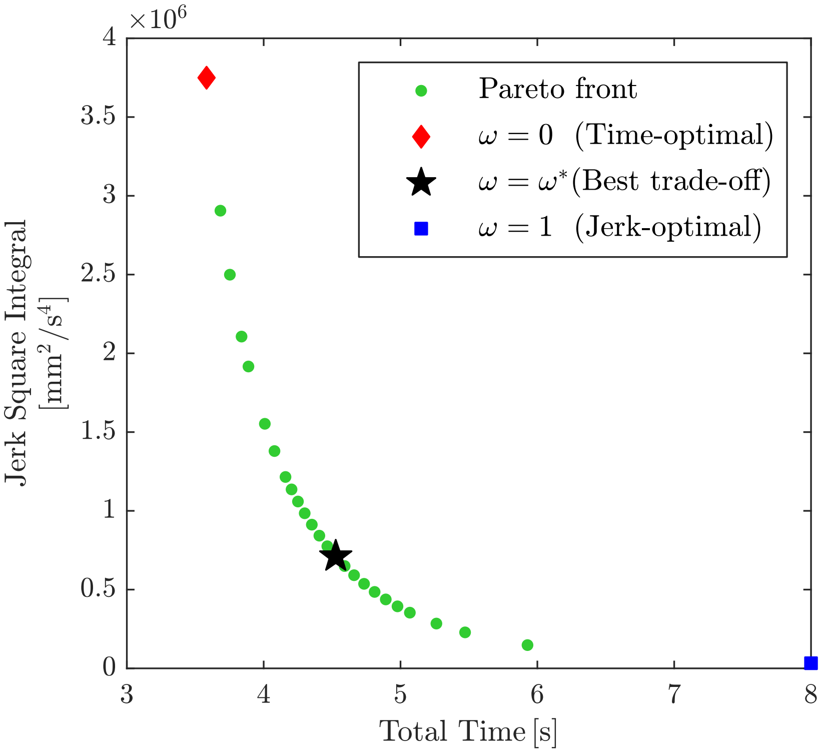


Fig. 3.9: Pareto front representing the optimization results of the GEMINI airfoil profile.

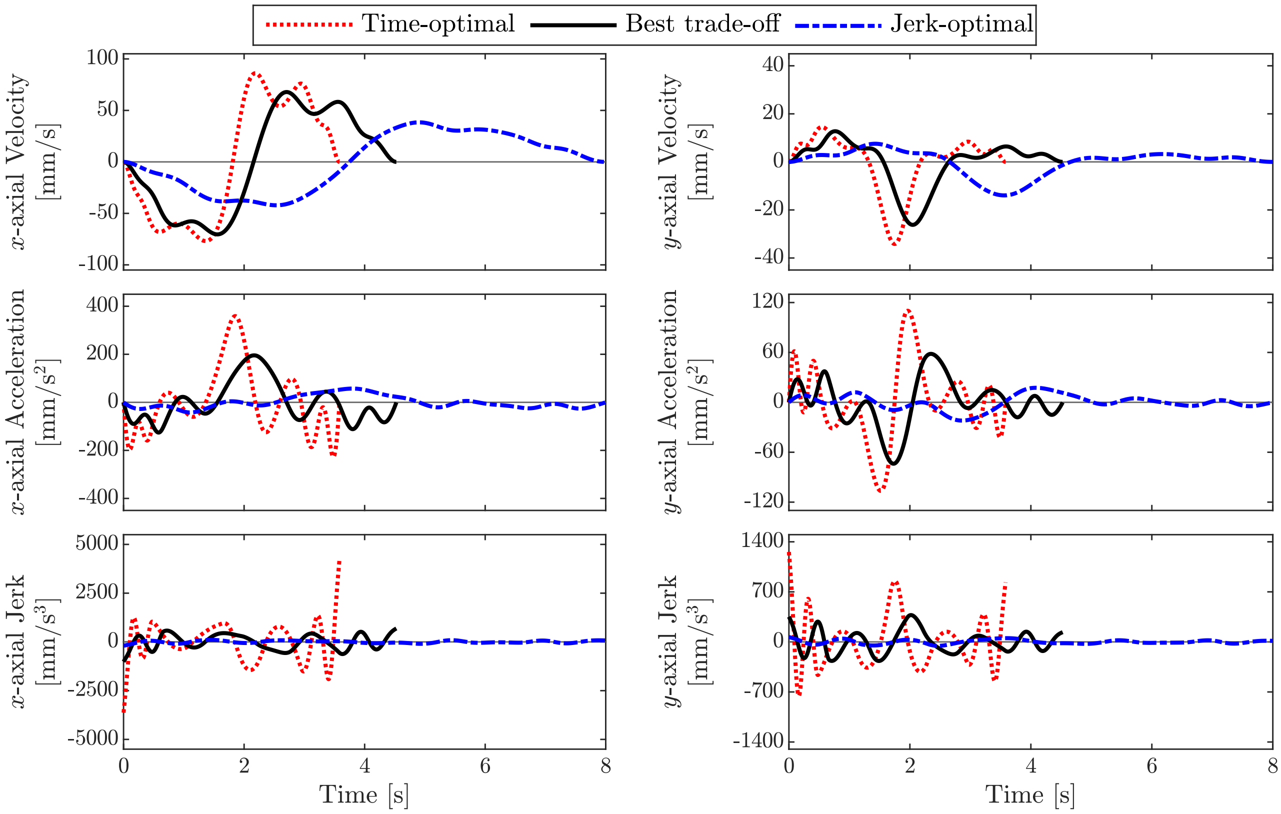


Fig. 3.10: GEMINI airfoil profile kinematic limits satisfaction for the x- and y-axis.

In linear reparameterization, the curve parameter is expressed as the constant scaling function of time as follows [7, 25]:

$$u = \lambda\, t, \quad t \in [0, t_{\mathrm{f}}], \tag{3.35}$$

where λ is a scaling factor of motion time t. In this case, the velocity, acceleration, and jerk vectors are obtained by multiplying the respective r^{th} derivatives of the parametric curve with λ^r. The value λ is defined as follows:

$$\lambda = \min\left\{ \frac{v_{\lim}}{|\dot{\boldsymbol{s}}(u)|_{\max}}, \quad \sqrt{\frac{a_{\lim}}{|\ddot{\boldsymbol{s}}(u)|_{\max}}}, \quad \sqrt[3]{\frac{j_{\lim}}{|\dddot{\boldsymbol{s}}(u)|_{\max}}} \right\}, \tag{3.36}$$

Eq. (3.36) is considered to assure maximum velocity, acceleration, and jerk limits; however, this method does not guarantee a smooth trajectory, which has the lowest possible jerk value, especially at the start and end. To achieve zero starting and ending velocities and accelerations,

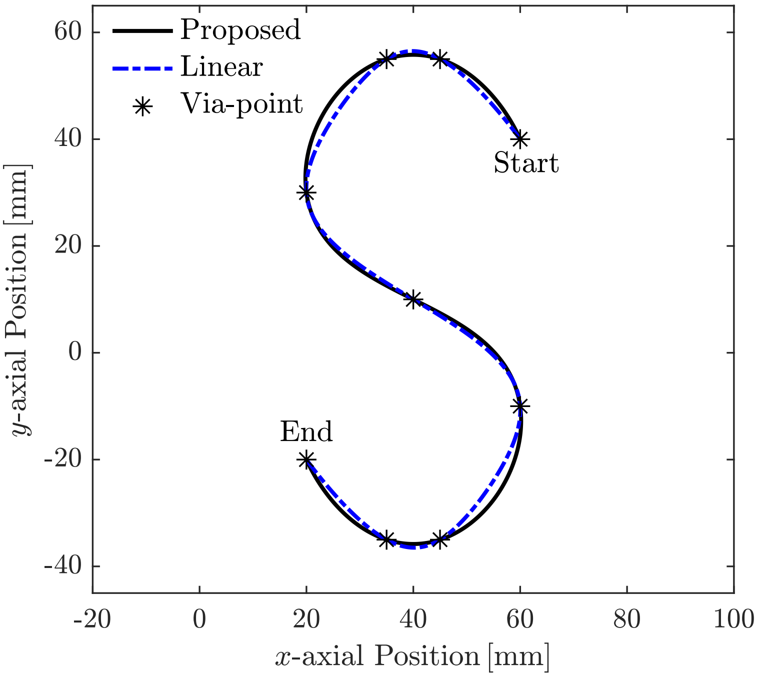


Fig. 3.11: Simulation results of the geometric path representation for proposed and linear reparameterization of the S-shaped profile.

the additional position coefficients and knots are required to impose the first and second derivatives of the geometric path to be zero at both ends such that the geometric path representation deviates from the proposed one, as shown in Fig. 3.11.

Fig. 3.12 shows the Pareto front representation for total time and jerk square integral of the proposed and linear reparameterization. All the trade-off solutions of the B-spline reparameterization are superior to those of the linear reparameterization. Compared with the best trade-off cases, the proposed method is approximately 3% faster, and the value of the jerk square integral is approximately 75% lesser than the linear reparameterization. Fig. 3.13 depicts a comparison of motion time, velocity, acceleration, and jerk for x- and y-axis of the respective best trade-off trajectories. By the proposed method, the peak values of jerk are reduced, especially at the start and the end of the trajectory, mainly due to the flexibility of the B-spline reparameterization function obtained by the optimization of the position coefficients and the total time. Besides, the boundary conditions for the zero velocities and accelerations are satisfied without changing the geometric shape, consequently leading to a smoother trajectory with a lower jerk value and faster time.

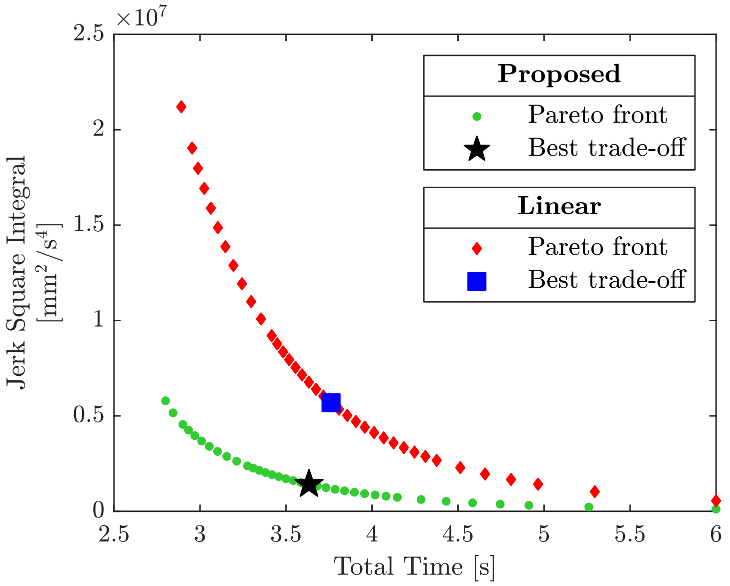


Fig. 3.12: Simulation results of the Pareto front (trade-off solutions between total time and jerk square integral) for proposed and linear reparameterization of the S-shaped profile.

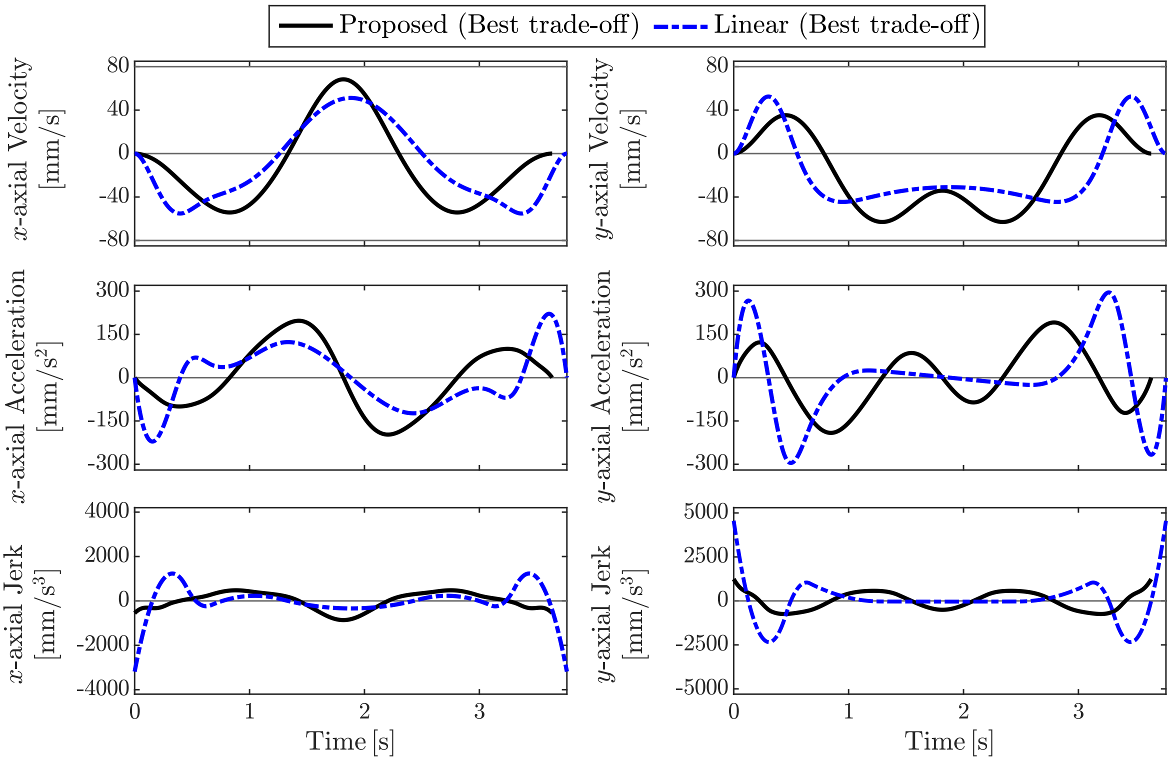


Fig. 3.13: Simulation results of motion time, velocity, acceleration, and jerk in x- and y-axis for proposed and linear reparameterization of the S-shaped profile.

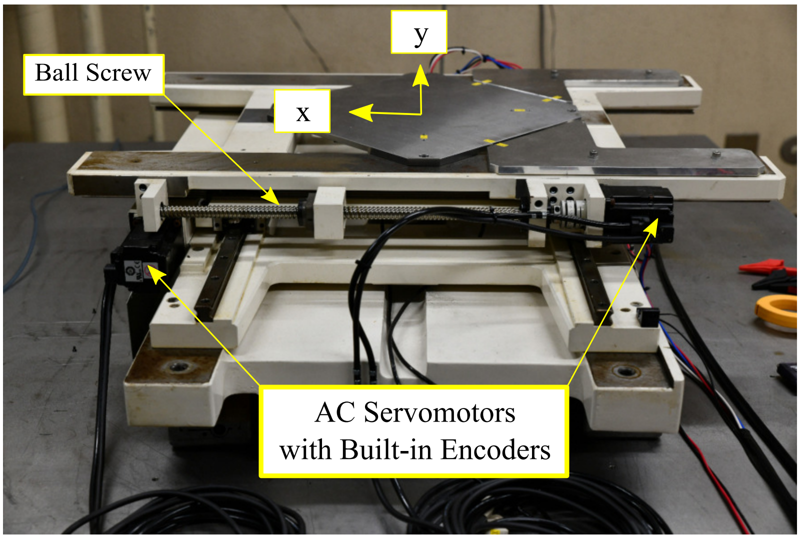


Fig. 3.14: Industrial biaxial feed drive system.

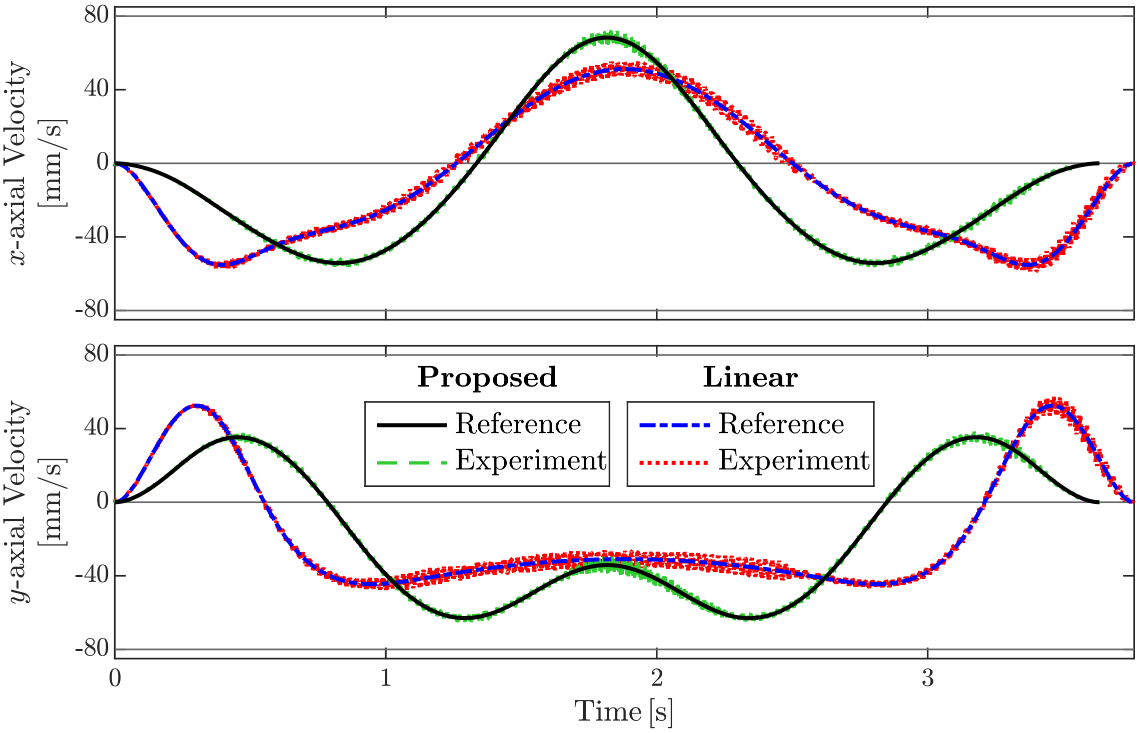


Fig. 3.15: Experimental verification of x- and y-axial velocities with reference (simulation) values for proposed and linear reparameterization of the S-shaped profile.

3.5.5 Experimental validation

To validate the performance of the proposed method, experiments were conducted with an industrial biaxial feed drive system shown in Fig. 3.14. It consists of two Alternating Current (AC) servomotors which transmit the motion in x- and y-directions via ball screws. The axial limits of the feed drive system are set to $\boldsymbol{v}_{\text{lim}} = [80, 80]$ mm/s, $\boldsymbol{a}_{\text{lim}} = [500, 500]$ mm/s^2, and $\boldsymbol{j}_{\text{lim}} = [20000, 20000]$ mm/s^3. The best trade-off trajectories of linear and proposed reparameterization compared in Section 3.5.4 were imported as reference trajectories to the feed drive system via a desktop computer of Intel(R) Core i7-3770K CPU, 3.50 GHz, 8GB RAM, and Ubuntu 15.04 64-bit operating system in a Xenomai 3.0 real-time framework. The conventional Proportional-derivative (PD) controller with viscous friction compensation was used to track the reference trajectories with the proportional and derivative gains of $10000\,\text{s}^{-2}$ and $200\,\text{s}^{-1}$, respectively. Position measurements were obtained with rotary encoders of 76.29 nm resolution at a sampling time of 0.2 ms.

The experiments were conducted seven times to guarantee the repeatability of the results. Velocity estimation was conducted by numerical differentiation of position measurements. Fig. 3.15 depicts the velocities for x- and y-axis of the linear and proposed method compared to their reference values. It was observed that the reference trajectories were trackable by the control system, and the time of the proposed method was faster than the linear reparameterization. Since the acceleration and jerk estimations were noisy, a comparison of tracking error was conducted for both cases instead because the lower jerk trajectory produced the smaller tracking errors [41]. By observing the results in Fig. 3.16, it is obvious that the tracking errors of the proposed method is lower than those of the linear reparameterization for both axes. Fig. 3.17 provides the mean absolute tracking errors in x- and y-axis of linear and proposed reparameterization for each iterative result. Due to the lower jerk trajectory of the proposed method, the mean absolute tracking errors in the x- and y-axis were reduced approximately 17.62% and 25.52%, respectively. Moreover, variations of the x- and y-axial tracking errors from the mean were approximately 17.37% and 25.66% smaller than the linear reparameterization. Therefore, these results validated that the proposed trajectory generation with kinematically constrained B-spline reparameterization provided better motion conditions in both time and jerk of industrial machines.

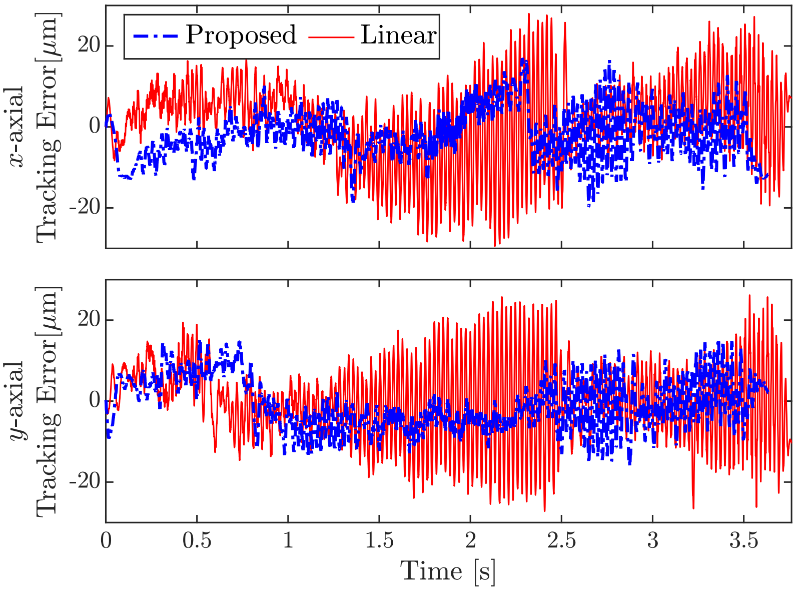


Fig. 3.16: Experimental tracking errors in x- and y-axis for linear and proposed reparameterization of the S-shaped profile.

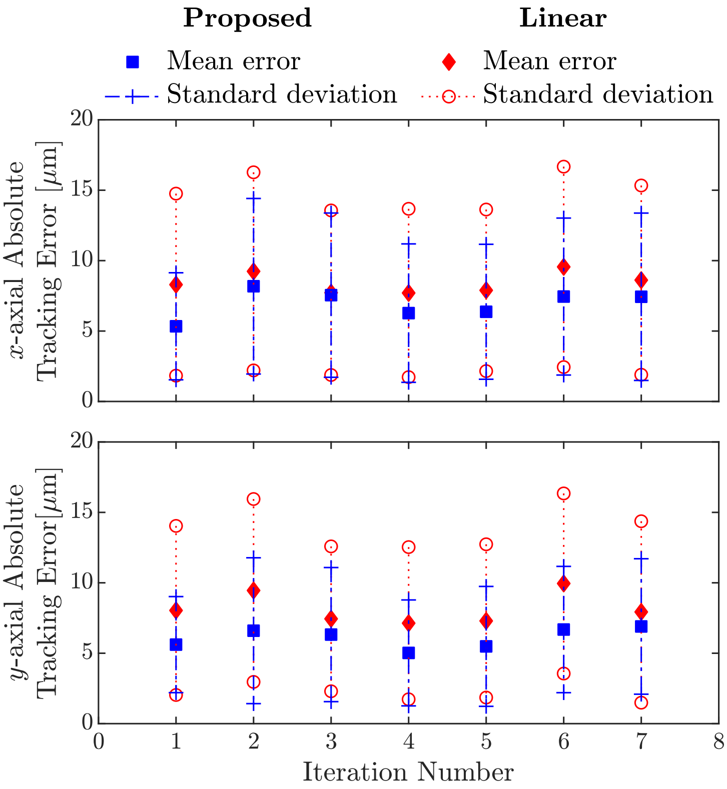


Fig. 3.17: Experimental mean absolute tracking error and standard deviation in x- and y-axis for linear and proposed reparameterization of the S-shaped profile.

3.6 Summary

This chapter proposes a decoupled spline-based approach for kinematically constrained reparameterization. The sixth-order parametric B-spline was used to represent the given via-points as a predefined geometric path, and the sixth-order B-spline reparameterization function was adopted for the optimization without changing the geometric shape. This technique considered both time and jerk minimization of the trajectory by the reparameterization function modification. The kinematic constraints were proposed herein based on the maximum geometric derivatives and the convex polygon formed by the pseudo-velocity, -acceleration, and -jerk control points. Therefore, the optimal solution guarantees the satisfaction of the kinematic limits for all times.

The Pareto front comprising significant trade-off solutions between total time and jerk square integral was revealed by the NNC method and the D&C algorithm, and each solution was computed by the SQP. The best trade-off solution was chosen subsequently. The proposed method was investigated with different geometric paths and limits, and simulations were implemented herein. The comparison with the linear reparameterization proved that the proposed method generated a smoother trajectory with a 75% lower jerk value and 3% faster time without changing the geometric shape. Due to a smoother trajectory, experiments results with an industrial FDS validates that the proposed method reduces the tracking errors by approximately 22% smaller than the linear reparameterization method. Therefore, the proposed optimal trajectory generation considering time and jerk is practically applicable to industrial machines.

Chapter 4

Smooth Time-Optimal Trajectory Generation with Guaranteed Kinematic Constraints

In this chapter, a decoupled approach for smooth and time-optimal trajectory generation of a predefined geometric path for CNC machines is proposed in the parameter domain. Chapter 3 presented a BOOP in the time domain, which considered the total motion time as one of the optimization variables, and the optimal time for each piecewise segment was achieved by subdividing the optimized total motion time equivalently. The kinematic constraints were determined based on the constant values of the highest geometric derivatives. As a drawback, the generated trajectory could not utilize the maximum ability of machine limits if the geometric path was complex (i.e., derivative values are frequently changed along the path). In other words, the conservatism of kinematic constraints was high. For those reasons, this chapter focuses on time-OCP in the parameter domain, in which time dependency is excluded in the problem formulation. A strategy for reducing the conservatism of kinematic constraints is proposed, and the comparative study is made with a jerk-limited STOTG method by simulations and experiments. This approach can be extended to the BOOP for considering a trade-off solution between motion time and jerk of the trajectories as in Chapter 3.

In this chapter, the proposed method uses the nonlinear transformation of variables [90], and time dependency is excluded in the problem formulation; therefore, time-optimal solutions for each piecewise segment are independently achieved. The cubic B-spline parameterizes the motion trajectory for acceleration and jerk continuity. Here, the velocity, acceleration, and jerk constraints are proposed based on the state control vectors and the maximum variable geometric derivatives in the locally affected parameter interval. As a result, the proposed method applies to complex geometric paths, and the resulting trajectory satisfies the kinematic constraints along the entire horizon. The control points of the pseudo-velocity square are considered as optimization variables, and the optimal solution is computed by SQP [84, 91]. Since comparing the results with our previous study (time-jerk optimal) [92] is difficult due to the different adoptions of an OCP, the results are compared with the related study [35], which is the jerk-constrained smooth and time-optimal trajectory generation in the parameter domain.

The rest of this chapter is organized as follows. Section 4.1 explains the related works concerning the proposed method. Section 4.2 briefly describes the formulations of time-optimal trajectories in previous studies. Section 4.3 adopts the control parameterization by the cubic B-spline. Section 4.4 explains the proposed approach for kinematic constraint satisfaction, followed by the reformulation of the smooth and time-optimal trajectory. Finally, Section 4.5 discusses the effectiveness of the proposed method by simulation and experimental results with different grid sizes, kinematic limits, and geometric paths, and Section 4.6 summarizes this chapter.

4.1 Related works

In Section 1.2.3.1, the literature review has been made on the time-optimal trajectory generations of CNC machines and robotic manipulators. Regarding a convex optimization approach, Verscheure *et al.* [90, 93] proposed a nonlinear transformation of variables to generate time-optimal trajectories in the parameter domain considering upper and lower bounds on the velocity, acceleration, and torque. Unlike [31], time variable was not explicitly included in the formulation of an OCP. However, considering only acceleration or torque constraints in an OCP may induce the vibration of the machine tool and large tracking errors [32]. Therefore, advanced controllers such as iterative learning [94], sliding mode [95], and model predictive control [9] were considered for compensating the errors.

An alternative approach for generating a smooth and time-optimal trajectory is the addition of jerk constraints in the trajectory formulation. In [38], the cubic spline was used to parameterize the motion trajectory along a specified path, and the smooth and time-optimal trajectory was formulated satisfying the pseudo-constraints. In [96, 97], the optimality condition was stated as follows: the time-optimal feed rate is maximum for all parameters; that is, at least one of the velocity, acceleration, or jerk limits is active (reach the maximum limits) at almost all points along the trajectory. In [35], Zhang *et al.* introduced an additional state for the jerk constraints, formulated a time-OCP considering pseudo and axial jerks as optimization variables, and proved that the active kinematic constraints are approximately bang–bang throughout the motion.

Regarding the abovementioned approaches, the solutions of time-optimal trajectories were mostly generated using the direct transcription method [54, 98], where the problem is discretized over the time or parameter horizon, and is solved numerically using nonlinear programming techniques. The constraints are imposed at specific grid points of the horizon. A user can determine the number of grids. For complex geometric paths, the derivatives are frequently changed over the parameter horizon. To satisfy the velocity, acceleration, and jerk limits, up to the third derivative of the path must be considered as kinematic constraints in the time minimization. Therefore, the drawback is that the violation of kinematic constraints may occur in-between the determined grid points when the number of grids is small. To overcome this problem, algorithms such as the pointwise discretization [99] can be used, where the problem number of grids is increased iteratively until the constraints are satisfied. However, this is not straightforward for the users. Increasing the number of grids introduces additional variables and constraints, thus inducing a more complicated problem. Hence, conservative approaches are introduced using the convex hull property by spline parameterization.

Concerning coupled conservative approaches, Gasparetto *et al.* [7, 45, 46] proposed a method for bi-objective time-jerk trajectory generation, satisfying kinematic constraints by the convex hulls of B-splines. Similarly, Mercy *et al.* [30] proposed conservative constraints for minimizing the motion time of CNC machines considering the accuracy of the generated workpiece. The optimal solution for the nonlinear systems with a guaranteed constraint satisfaction was proposed based on the convex hull of splines [100]. For the decoupled conservative approach, Paing *et al.* [92] represented a given intermediate points by the sixth-order B-splines and generated a time-jerk optimal trajectory considering the kinematic constraints in the time domain.

4.2 Problem formulation

The geometric path of the workpiece represents the three-dimensional position of the tool path for CNC machines, which is usually provided in the parametric form $\boldsymbol{s}(u) = [X(u), Y(u), Z(u)]^T$, where $u \in [0, 1]$ is the curve parameter. If the predefined geometric path is second-order continuous or higher, the velocity, acceleration, and jerk in time horizon are defined as in (3.3)-(3.5).

To generate time-optimal trajectories, the minimum total motion time, which satisfies the upper and lower kinematic limits of machines, must be considered. The initial and final boundary conditions for velocity and acceleration are added for the smooth transition at the start and end of the motion trajectory. Therefore, the original OCP for minimizing the total motion time is described as follows [97]:

$$\min_{\dot{u}(t), \ddot{u}(t), \dddot{u}(t)} \int_0^{t_\mathrm{f}} 1\, dt, \tag{4.1}$$

subject to

$$u(0) = 0, \;\; u(t_\mathrm{f}) = 1, \tag{4.2}$$

$$\tilde{\boldsymbol{v}}(0) = \boldsymbol{v}_0, \;\; \tilde{\boldsymbol{v}}(t_\mathrm{f}) = \boldsymbol{v}_\mathrm{f}, \tag{4.3}$$

$$\tilde{\boldsymbol{a}}(0) = \boldsymbol{a}_0, \;\; \tilde{\boldsymbol{a}}(t_\mathrm{f}) = \boldsymbol{a}_\mathrm{f}, \tag{4.4}$$

$$\boldsymbol{v}_\mathrm{min} \leqslant \boldsymbol{s}'\, \dot{u} \leqslant \boldsymbol{v}_\mathrm{max}\,, \tag{4.5}$$

$$\boldsymbol{a}_\mathrm{min} \leqslant \boldsymbol{s}''\, \dot{u}^2 + \boldsymbol{s}'\, \ddot{u} \leqslant \boldsymbol{a}_\mathrm{max}\,, \tag{4.6}$$

$$\boldsymbol{j}_\mathrm{min} \leqslant \boldsymbol{s}'''\, \dot{u}^3 + 3\, \boldsymbol{s}''\, \ddot{u}\, \dot{u} + \boldsymbol{s}'\, \dddot{u} \leqslant \boldsymbol{j}_\mathrm{max}\,. \tag{4.7}$$

where $\boldsymbol{v}_0$ and $\boldsymbol{v}_f$ are the initial and final velocities, and $\boldsymbol{a}_0$ and $\boldsymbol{a}_f$ are the initial and final accelerations, respectively. The nonlinear transformation of variables for velocity and acceleration are used; therefore, the states are determined as follows [90, 93]:

$$\alpha(u) = \dot{u}^2,\; \beta(u) = \ddot{u}, \tag{4.8}$$

and their derivatives with respect to parameter u:

$$\alpha'(u) = 2\,\ddot{u},\ \beta'(u) = \frac{1}{2}\alpha''(u). \tag{4.9}$$

To consider the jerk constraints, the additional state is defined as follows [35]:

$$\gamma(u) = \frac{\dddot{u}}{\dot{u}} = \beta'(u), \tag{4.10}$$

By defining (4.8)–(4.10), the objective function in (4.1) can be rewritten by changing the integration variable from t to u as follows:

$$T = \int_0^{t_\mathrm{f}} 1\,dt = \int_{u(0)}^{u(t_\mathrm{f})} \frac{1}{\sqrt{\alpha(u)}}\,du = \int_0^1 \frac{1}{\sqrt{\alpha(u)}}\,du. \tag{4.11}$$

If the upper and lower kinematic limits in (4.5)–(4.7) are assumed to be symmetric in all axes, the OCP for minimizing the total motion time can be reformulated in the parameter domain, defined as follows [35]:

$$\min_{\alpha(u)} \int_0^1 \frac{1}{\sqrt{\alpha(u)}}\,du, \tag{4.12}$$

subject to

$$\alpha(u) > 0,\ \alpha'(u) = 2\,\beta(u),\ \gamma(u) = \beta'(u), \tag{4.13}$$

$$\alpha(0) = \alpha_0,\ \alpha(1) = \alpha_\mathrm{f}, \tag{4.14}$$

$$\beta(0) = \beta_0,\ \beta(1) = \beta_\mathrm{f}, \tag{4.15}$$

$$[\boldsymbol{s}']^2\,\alpha(u) \leqslant \boldsymbol{v}_\mathrm{lim}^2, \tag{4.16}$$

$$|\boldsymbol{s}''\,\alpha(u) + \boldsymbol{s}'\,\beta(u)| \leqslant \boldsymbol{a}_\mathrm{lim}, \tag{4.17}$$

$$|\boldsymbol{s}'''\,\alpha(u) + 3\ \boldsymbol{s}''\ \beta(u) + \boldsymbol{s}'\ \gamma(u)|\ \sqrt{\alpha(u)} \leqslant \boldsymbol{j}_\mathrm{lim}. \tag{4.18}$$

where α_0 and α_f represent the initial and final state values of $\alpha(u)$, β_0 and β_f are the initial and final state values of $\beta(u)$, respectively. The initial and final velocity and acceleration of the trajectory are satisfied by the equality constraints of the states in (4.14) and (4.15). Thereafter, the time-optimal solution for (4.12) is formulated by imposing a finite set of inequality kinematic constraints in (4.16)–(4.18) at specific grid points of the trajectory.

4.3 Control parameterization by cubic B-splines

In this study, a direct transcription method is used to solve the OCP described in (4.12)–(4.18); therefore, grid points $\{0, 1, 2, 3, \ldots, N\}$ are determined to discretize the problem in $u \in [0, 1]$ as follows [35, 90, 93]:

$$0 = d_0 < d_1 < d_2 < \ldots < d_{N-1} < d_N = 1. \tag{4.19}$$

To establish continuity in the acceleration and jerk of a given second- or third-order continuous geometric path, the state $\alpha(u)$ should be cubic or higher degrees. Here, the cubic B-spline function with the order $(q = 4)$ comprising of $N + 3$ control points, which makes N piecewise polynomial segments, is used to parameterize the motion trajectory as follows [25]:

$$\alpha(u) = \sum_{i=0}^{N+2} B_{i,q}(u)\, c_{\alpha,i},\ 0 \leqslant u \leqslant 1, \tag{4.20}$$

with a non-decreasing knot vector

$$\hat{u} = [\,\underbrace{0, \ldots, 0}_{q\text{-times}}, d_1, d_2, \ldots, d_{N-1}, \underbrace{1, \ldots, 1}_{q\text{-times}}\,], \tag{4.21}$$

where $B_{i,q}(u)$ is the basis function, and $\hat{\boldsymbol{c}}_\alpha = [c_{\alpha,0}, c_{\alpha,1}, \ldots, c_{\alpha,N+2}]^T$ represents the control input vector in this study. The knot vector in (4.21) is q-times repeated at the start and end; therefore, the initial and final states values $\alpha(0)$ and $\alpha(1)$ coincide with the control inputs $c_{\alpha,0}$ and $c_{\alpha,N+2}$, respectively. For simplicity, the knots in (4.21) are not considered as optimization variables but are assumed to be uniformly distributed over the parameter interval $[0, 1]$. Using (4.8)–(4.10), the derivatives of the state $\beta(u)$ and $\gamma(u)$ are determined as follows:

$$\beta(u) = \frac{1}{2} \sum_{i=0}^{N+2} B'_{i,q}(u)\, c_{\alpha,i}. \tag{4.22}$$

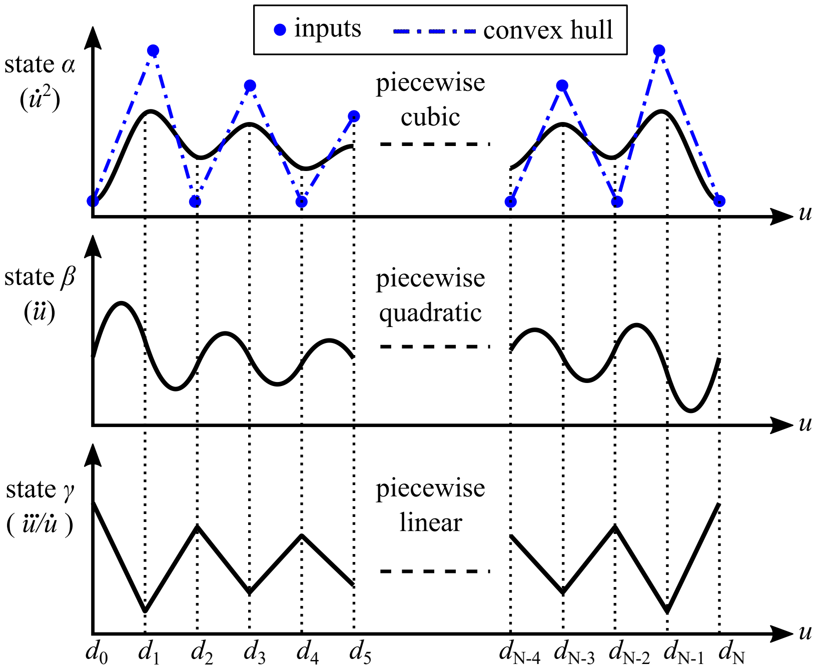


Fig. 4.1: Control parameterization of the states α, β, and γ and the convex hull comprising the inputs (control points) of the proposed method.

$$\gamma(u) = \frac{1}{2} \sum_{i=0}^{N+2} B''_{i,q}(u)\, c_{\alpha,i}. \tag{4.23}$$

Fig. 4.1 shows the control parameterization of the states. Since the state α is piecewise cubic, the states β and γ are piecewise quadratic and piecewise linear, respectively.

4.4 Guaranteed kinematic constraint approach

In Section 4.3, the inputs are defined as the control points $\hat{\boldsymbol{c}}_\alpha$, and the states are the piecewise continuous functions $\alpha(u)$, $\beta(u)$, and $\gamma(u)$; therefore, the outputs are the velocity, acceleration, and jerk of the trajectory, which must satisfy the kinematic limits along the parameter interval $u \in [0, 1]$. Here, the kinematic constraints are proposed based on the control points of the states, which are known as the convex hull points. The convex hull property of the B-splines states that a spline function must exist within the convex hull formed by its control points [11].

Therefore, we determine the control points of the respective states as follows [7]:

$$\hat{\boldsymbol{c}}_\beta = [c_{\beta,0}, c_{\beta,1}, \ldots, c_{\beta,N+1}]^T, \tag{4.24}$$

with

$$c_{\beta,i} = \frac{(q-1)}{2(u_{i+q} - u_{i+1})}\,(c_{\alpha,i+1} - c_{\alpha,i}), i = \{0, 1, \ldots, N+1\}. \tag{4.25}$$

Similarly, it is defined that

$$\hat{\boldsymbol{c}}_\gamma = [c_{\gamma,0},\ c_{\gamma,1},\ \ldots,\ c_{\gamma,N}]^T, \tag{4.26}$$

with

$$c_{\gamma,i} = \frac{(q-2)}{2(u_{i+q} - u_{i+2})}\,(c_{\beta,i+1} - c_{\beta,i}), i = \{0, 1, \ldots, N\}. \tag{4.27}$$

where $\hat{\boldsymbol{c}}_\beta$ and $\hat{\boldsymbol{c}}_\gamma$ are the control vectors of the states $\beta(u)$ and $\gamma(u)$, respectively. According to the local modification scheme of B-splines [11], the change of input $c_{\alpha,i}$ affects only the parameterization function $\alpha(u)$ in the interval $[u_i, u_{i+q})$; therefore, we change only the highest geometric derivatives in the locally affected intervals for the proposed constraints. The velocity constraint in (4.16) is substituted with a finite set of proposed constraints as follows:

$$\tilde{v}_i \leqslant v_{\text{lim}}^2,\ i = \{0, 1, \ldots, N+2\}, \tag{4.28}$$

with

$$\tilde{v}_i = \lambda_{1,i}^2\ \ c_{\alpha,i}, \tag{4.29}$$

$$\lambda_{1,i} = \max\{\max|\boldsymbol{s}'(u)|\}, u \in [u_i, u_{i+q}), \tag{4.30}$$

where $\lambda_{1,i}$ is the i^{th} highest absolute scalar first derivative of the predefined geometric path between the locally affected interval by the control input $c_{\alpha,i}$.

The example is illustrated in Fig. 4.2 with the control inputs $c_{\alpha,0}$ and $c_{\alpha,1}$. Due to the repeated knots at the start, the input $c_{\alpha,0}$ locally affects the function $\alpha(u), u \in [d_0, d_1)$, whereas the input $c_{\alpha,1}$ affects $u \in [d_0, d_2)$. Therefore, to satisfy the velocity constraints, the highest first geometric

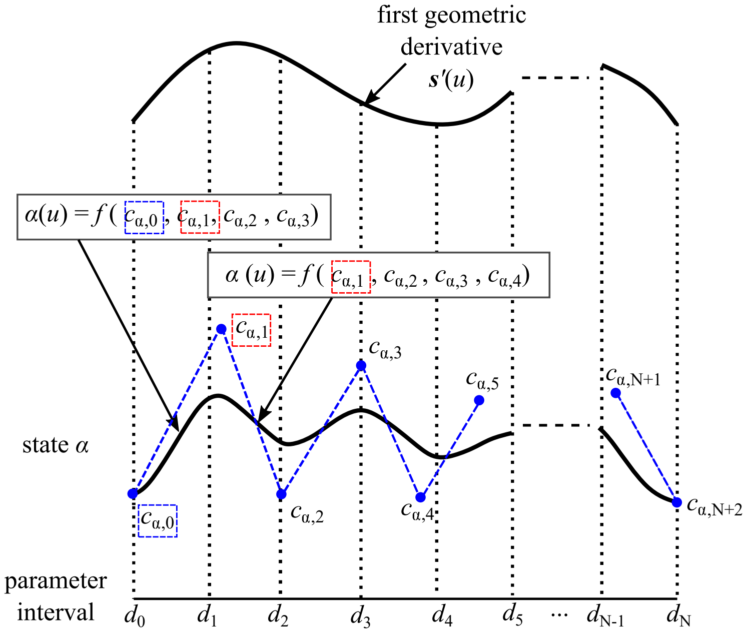

first geometric derivative
s′(u)
α(u) = f(cα,0 , cα,1, cα,2 , cα,3)
α (u) = f(cα,1 , cα,2 , cα,3 , cα,4)
cα,1
cα,3
cα,5
cα,N+1
state α
cα,0
cα,2
cα,4
cα,N+2
parameter interval
d0
d1
d2
d3
d4
d5
···
dN-1
dN

Fig. 4.2: An example for choosing the control inputs and the first derivatives of the predefined geometric path for the proposed velocity constraint.

derivatives $\lambda_{1,0}(u), u \in [d_0, d_1)$ and $\lambda_{1,1}(u), u \in [d_0, d_2)$ are changed together with the control inputs $c_{\alpha,0}$ and $c_{\alpha,1}$, respectively.

According to (4.25), $c_{\beta,i}$ is the rate of change of the inputs $c_{\alpha,i}$ and $c_{\alpha,i+1}$. To satisfy the acceleration constraints in (4.17), two acceleration constraints are checked for each i^{th} iteration as follows:

$$\tilde{a}_{i,k} \leqslant a_{\text{lim}},\ i = \{0, 1, \ldots, N+1\}, k = \{0, 1\}, \tag{4.31}$$

with

$$\tilde{a}_{i,k} = \varphi_{2,i}\ \ c_{\alpha,i+k} + \varphi_{1,i}\ \ |c_{\beta,i}|, \tag{4.32}$$

$$\varphi_{1,i} = \max\{\max|\boldsymbol{s}'(u)|\}, u \in [u_i, u_{i+q+1}), \tag{4.33}$$

$$\varphi_{2,i} = \max\{\max|\boldsymbol{s}''(u)|\}, u \in [u_i, u_{i+q+1}), \tag{4.34}$$

where $\varphi_{1,i}$ and $\varphi_{2,i}$ are the updated i^{th} absolute scalar first and second derivatives of the geometric path, which demonstrate the highest values between the locally affected interval by the inputs $c_{\alpha,i}$ and $c_{\alpha,i+1}$.

To generate smooth trajectories, jerk constraints are proposed in this study. Considering (4.27), $c_{\gamma,i}$ is the rate of change of $c_{\beta,i}(c_{\alpha,i}, c_{\alpha,i+1})$ and $c_{\beta,i+1}(c_{\alpha,i+1}, c_{\alpha,i+2})$. To satisfy (4.18), four jerk constraints are checked for each i^{th} iteration, defined as follows:

$$\tilde{j}_{i,k} \leqslant j_{\max},\ i = \{0, 1, \ldots, N\}, k = \{0, 1, 2, 3\}, \tag{4.35}$$

with

$$\tilde{j}_{i,k} = \begin{cases} \sqrt{c_{\alpha,i+k}}(\psi_{3,i}\ c_{\alpha,i+k} + 3\,\psi_{2,i}\ |c_{\beta,i}| \\ +\,\psi_{1,i}\,|c_{\gamma,i}|), \text{ for } k \leqslant 1, \\ \sqrt{c_{\alpha,i+k-1}}(\psi_{3,i}\ c_{\alpha,i+k-1} + 3\,\psi_{2,i}\ |c_{\beta,i+1}| \\ +\,\psi_{1,i}\,|c_{\gamma,i}|), \text{ otherwise.} \end{cases} \tag{4.36}$$

$$\psi_{1,i} = \max\{\max|\boldsymbol{s}'(u)|\}, u \in [u_i, u_{i+q+2}), \tag{4.37}$$

$$\psi_{2,i} = \max\{\max|\boldsymbol{s}''(u)|\}, u \in [u_i, u_{i+q+2}), \tag{4.38}$$

$$\psi_{3,i} = \max\{\max|\boldsymbol{s}'''(u)|\}, u \in [u_i, u_{i+q+2}), \tag{4.39}$$

where $\psi_{1,i}$, $\psi_{2,i}$, and $\psi_{3,i}$ are the updated i^{th} absolute scalar first, second, and third derivatives of the geometric path, which are the highest between the locally affected interval by the inputs $c_{\alpha,i}$, $c_{\alpha,i+1}$, and $c_{\alpha,i+2}$. The lower bounds of the inputs are greater than zero, and the proposed kinematic constraints determine the upper bounds of the inputs. The pseudo-code for calculating kinematic constraints of (4.28)–(4.39) is provided in Algorithm 1.

Moreover, the initial and final boundary values of the velocity and acceleration must be determined to achieve the smooth start and end transitions of the trajectory. According to B-splines property, the first and last control points coincide with their respective state values at 0 and 1; therefore, the equality constraints are determined as follows:

$$c_{\alpha,0} = \alpha_0,\ \ c_{\alpha,N+2} = \alpha_{\text{f}}. \tag{4.40}$$

$$c_{\beta,0} = \beta_0,\ \ c_{\beta,N+1} = \beta_{\text{f}}. \tag{4.41}$$

To efficiently estimate the objective function in (4.12), the midpoints of (4.21) are calculated as follows:

$$\tilde{d}_i = \begin{cases} 0, & \text{for } i = 0, \\ \frac{1}{2}\left(d_{i-1} + d_i\right), & \text{for } i = \{1, 2, \ldots, N\}, \\ 1, & \text{for } i = N + 1. \end{cases} \tag{4.42}$$

Therefore, the proposed smooth and time-optimal OCP with velocity, acceleration, and jerk constraints can be formulated as follows:

$$\min_{\hat{\boldsymbol{c}}_\alpha} \sum_{i=1}^{N+1} \frac{(\tilde{d}_i - \tilde{d}_{i-1})}{\sqrt{\alpha(\tilde{d}_i)}}, \tag{4.43}$$

subject to $\hat{\boldsymbol{c}}_\alpha > 0$ and (4.28)–(4.41).

4.5 Application results

4.5.1 Calculation conditions

In this section, to evaluate the performance of the proposed method, time-optimal trajectories are computed for several geometric paths and kinematic limits. Different grid sizes are used to discretize the problem, and the OCPs are solved by the SQP (“fmincon” function) in a MATLAB® environment of a Core i7-7500U CPU and 8 GB RAM laptop computer with a Windows 10 64-bit operating system. Next, the resulting trajectories are compared with the STOTG, which uses discrete kinematic constraints on grid points for calculation [35]. We focus on the comparison of kinematic constraint satisfaction along the trajectory, total motion time, and computation time of the algorithms. Different grid sizes, kinematic limits, and geometric paths are used to formulate the OCPs to clarify the effectiveness of the proposed method. Also, the strategy for reducing the conservatism of kinematic constraints is described. The STOTG uses the pseudo and the axial jerks of the trajectory as optimization variables, whereas the proposed method uses state control points as optimization variables, which are the convex hull points of the pseudo-velocity square.

Algorithm 1 Calculation of proposed inequality constraints for velocity, acceleration, and jerk

```
Input: $\hat{\boldsymbol{c}}_\alpha$                                   ▷ control inputs
Output: $\boldsymbol{\Phi}$          ▷ A set containing proposed inequality kinematic constraints
procedure kinematicConstraints
    $\boldsymbol{\Phi} \leftarrow \{\}$
    for $i = 0;\quad i \leqslant N+2;\quad i++$ do
        Define the interval $[u_i, u_{i+q})$
        Calculate $\lambda_{1,i}$                                ▷ From (4.30)
        Calculate $\tilde{v}_i$                         ▷ From (4.28) and (4.29)
        $\boldsymbol{\Phi} \leftarrow \tilde{v}_i$
    end for
    for $i = 0;\quad i \leqslant N+1;\quad i++$ do
        Define the interval $[u_i, u_{i+q+1})$
        Calculate $\varphi_{1,i}$ and $\varphi_{2,i}$               ▷ From (4.33) and (4.34)
        for $k = 0;\quad k \leqslant 1;\quad k++$ do
            Calculate $\tilde{a}_{i,k}$                 ▷ From (4.31) and (4.32)
            $\boldsymbol{\Phi} \leftarrow \tilde{a}_{i,k}$
        end for
    end for
    for $i = 0;\quad i \leqslant N;\quad i++$ do
        Define the interval $[u_i, u_{i+q+2})$
        Calculate $\psi_{1,i}$, $\psi_{2,i}$, and $\psi_{3,i}$                ▷ From (4.37)–(4.39)
        for $k = 0;\quad k \leqslant 3;\quad k++$ do
            Calculate $\tilde{j}_{i,k}$                 ▷ From (4.35) and (4.36)
            $\boldsymbol{\Phi} \leftarrow \tilde{j}_{i,k}$
        end for
    end for
    Return $\boldsymbol{\Phi}$
end procedure
```

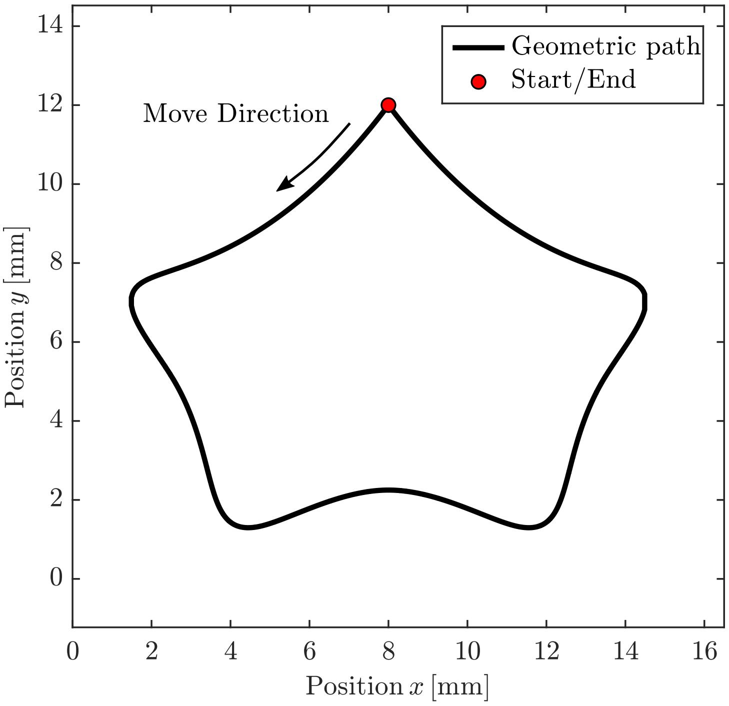


Fig. 4.3: Geometric positions of the star-shaped profile.

4.5.2 Application to the star-shaped geometric path

In the first example, the second-order continuous star-shaped profile is selected from [35], and Fig. 4.3 shows the given geometric profile. The absolute limits for velocity, acceleration, and jerk are defined as $v_{\text{lim}} = 80$ mm/s, $a_{\text{lim}} = 500$ mm/s^2, and $j_{\text{lim}} = 10000$ mm/s^3, respectively. The initial and final states of the trajectory are selected as $\alpha_0 = \alpha_{\text{f}} = 0.001\,\text{s}^{-1}$ and $\beta_0 = \beta_{\text{f}} = 0\,\text{s}^{-2}$. The grid size for solving the problem is chosen as $N = 30$.

Fig. 4.4 illustrates the comparison of velocity, acceleration, and jerk, including the grid point values for each axis of the proposed method and STOTG. Both trajectories satisfy the initial and final state values and achieve smooth transitions at the start and end of the trajectory. The constraints for velocity, acceleration, and jerk are satisfied at the grid points for both cases. Also, the optimal trajectories satisfy velocity and acceleration constraints along the predefined geometric path, although the STOTG violates the jerk constraint. This behavior occurs when only discrete constraints are considered on grid points; therefore, constraint violations may still occur between the grid points.

In the proposed method, kinematic constraints are applied based on the state control points (convex hull points) and the highest variable derivatives of the predefined geometric path in the locally affected parameter interval. Therefore, the proposed method guarantees the kinematic constraint satisfaction for all $u \in [0, 1]$. Fig. 4.5 illustrates the absolute velocities of both algorithms in parameter horizon $[0, 1]$. As the axial jerk value exceeds the limits, the absolute velocity of the STOTG becomes approximately 33.75% higher than that of the proposed method in-between the two grid points (highlighted in yellow). However, in the proposed method, the axial jerk values are bounded by the limits, thereby inducing a smoother velocity profile.

The motion times are obtained according to the objective functions of the respective OCPs. For a lower number of grids, the obtained motion time of the STOTG is slower than that of the proposed method. Since the parametric distance is large, the optimized pseudo-velocity is small, especially at the start and end of the trajectory. As a result, the total motion times of the proposed method and STOTG are 2.21 s and 2.31 s, respectively. The computation times for the proposed method and STOTG are 0.45 s and 0.8 s, respectively, for this problem. Therefore, the proposed method provides an approximately 4.32% faster trajectory and 43.75% lower computation time than the STOTG without violating the jerk constraints.

4.5.3 Application to the butterfly-shaped geometric path

In the second example, a more complex third-order continuous butterfly-shaped geometric path was predefined using the sixth-order B-splines with the 101 control points in the x- and y-directions. Fig. 4.6 shows the predefined geometric path. The symmetric axial limits for velocity, acceleration, and jerk for the trajectory are set as $v_{\text{lim}} = 80$ mm/s, $a_{\text{lim}} = 1000$ mm/s^2, and $j_{\text{lim}} = 20000$ mm/s^3, respectively. The initial and final states are chosen the same as the previous example in Section 4.5.2.

Different grid sizes of 50 and 150 are used to formulate the OCP, and the total motion time, computation time, and kinematic constraint satisfaction of the resulting trajectories are compared. Fig. 4.7 shows the axial velocity, acceleration, and jerk of both algorithms with a grid size of 50. In this case, the STOTG violates acceleration and jerk constraints between the determined grid points, whereas the proposed method guarantees constraint satisfaction of the

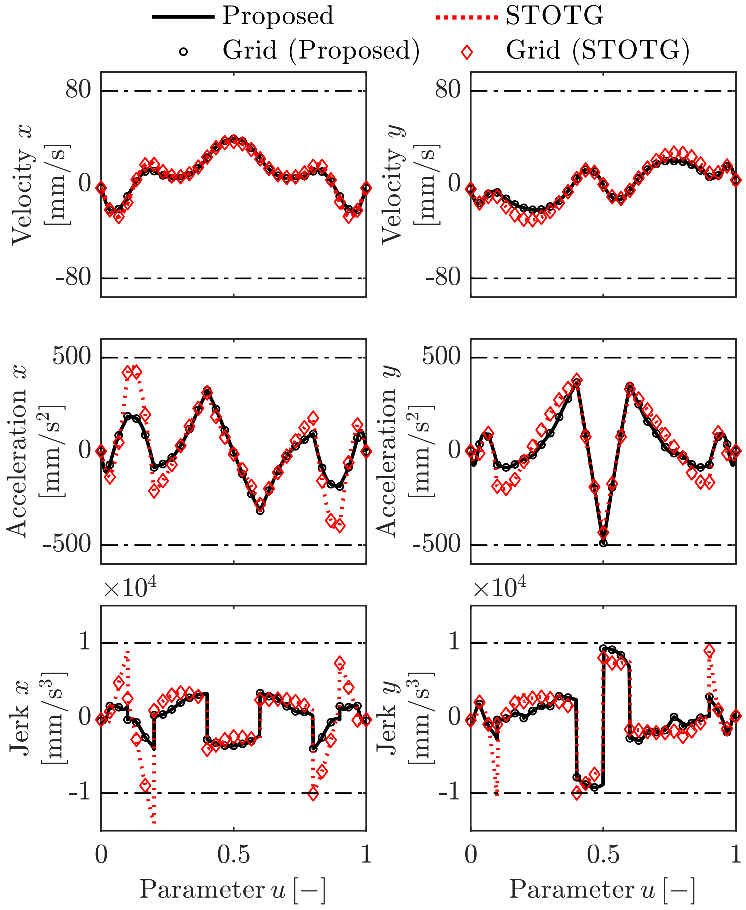


Fig. 4.4: Simulation results of velocity, acceleration, and jerk and the values at the grid points for each axis of the proposed method and STOTG with a grid size of $N = 30$.

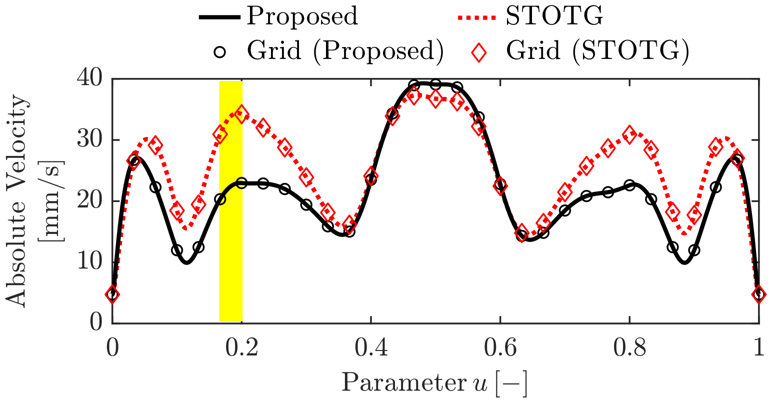


Fig. 4.5: The absolute velocities of the proposed method and STOTG for the star-shaped profile with a grid size of $N = 30$ (Simulation results).

Table 4.1: Comparison of the computation time (t_{comp}), total motion time (t_{f}), and kinematic constraint satisfaction for the proposed method and STOTG with grid sizes of $N = 50$ and $N = 150$ for the butterfly-shaped geometric path (Simulation results).

N	Method	t_{comp}	t_{f}	Kinematic constraints
50	Proposed	1.76 s	6.08 s	Satisfied
	STOTG	8.82 s	4.44 s	Not Satisfied (Acceleration and Jerk)
150	Proposed	85.6 s	5 s	Satisfied
	STOTG	206.85 s	4.45 s	Not Satisfied (Jerk)

trajectory for all parameter intervals. The kinematic values satisfy the maximum limits due to the conservatism of proposed constraints.

Fig. 4.8 shows the axial kinematic values along the trajectory with a grid size of 150 for the proposed method and STOTG. The finer discretization of the problem introduces more optimization variables and constraints, thus increasing the computation time of both algorithms. Table 4.1 provides the computation time, total motion time, and kinematic constraint satisfaction of the resulting trajectories according to the various grid sizes of the OCP. The computation times of the proposed method and STOTG increase approximately 49 times and 23 times, respectively, with an increased grid size of 150. Observably, the STOTG consumes a larger computation time than the proposed method for each grid size due to a larger number of optimization variables (i.e., pseudo and axial jerks of the trajectory). By comparing the results in Figs. 4.7 and 4.8, the kinematic constraint violations along the trajectory were reduced in the STOTG, especially in the axial acceleration and jerk because more discrete constraints were imposed along the trajectory. Therefore, the STOTG increases its accuracy of kinematic constraints when the number of grids is increased. Moreover, increasing optimization variables in STOTG makes the OCP more flexible, thus the motion times of 50 and 150 grids are similar. The proposed method is more robust to the number of grids for kinematic constraints. In other words, the proposed method guarantees the constraint accuracy between grid points.

Increasing the number of grids in the proposed method introduces more state control points, which are the optimization variables of the OCP, and become more flexible in handling different derivatives of the geometric path in the problem formulation. The resulting kinematic values become closer to the maximum limits (See Fig. 4.8); therefore, the conservatism of kinematic

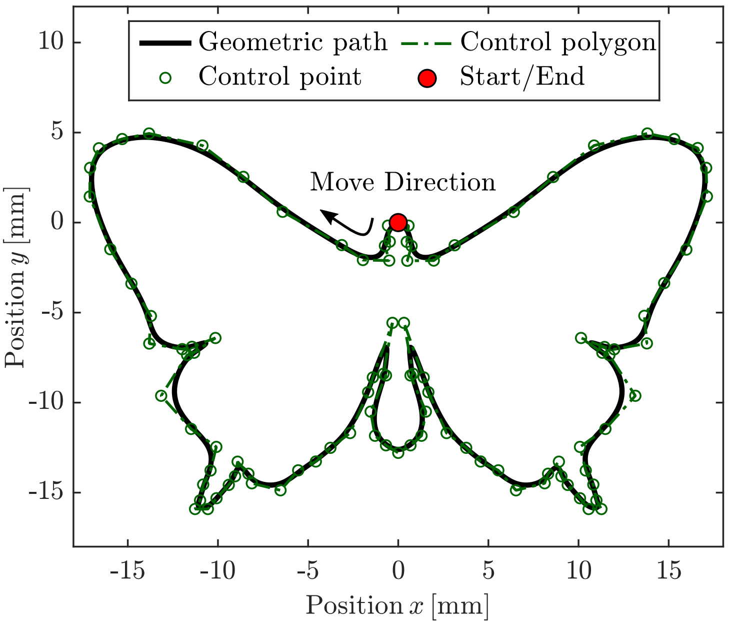


Fig. 4.6: Geometric path of the butterfly-shaped profile comprising 101 control points by the sixth-order B-splines.

constraints is reduced. According to the optimality condition [96, 97], at least one of the velocity, acceleration, or jerk constraints must be active (reach the limits) in time-optimal trajectories. By observing the results in Fig. 4.9, the proposed jerk constraints are active at almost all points along the trajectory. For each iteration, one of the combinations of the jerk values in (4.36) reaches the absolute maximum limit. Therefore, the proposed trajectory satisfies the optimality condition and is time-optimal. Moreover, it also guarantees the jerk continuity for the third-order continuous geometric path.

Fig. 4.10 shows the absolute velocities of the proposed method and the STOTG with 50 and 150 grids. The proposed method exhibits a smoother velocity profile to satisfy the kinematic limits for all horizons. For the STOTG, the maximum absolute velocities with 150 grids become lower than those of 50 grids because constraint violation is reduced in the axial acceleration and jerk values. For the proposed method, the absolute velocities with an increased number of grids become higher owing to the reduced conservatism of the algorithm. Hence, the total motion time of the proposed method with 150 grids is approximately 17.72% faster than that of 50 grids.

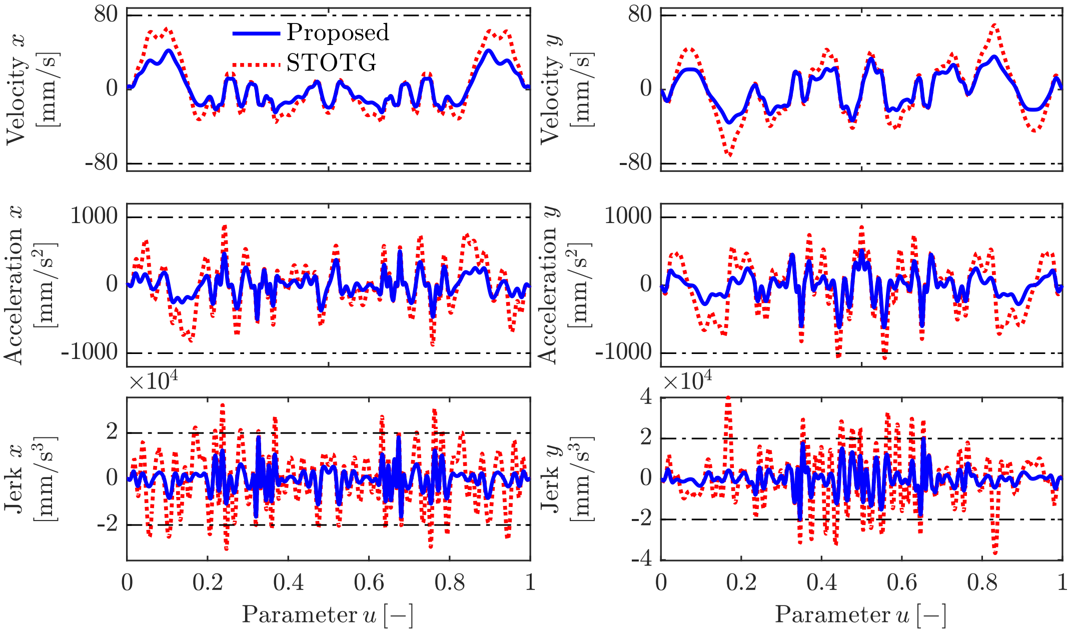


Fig. 4.7: Simulation results of velocity, acceleration, and jerk for each axis of the proposed method and STOTG with a grid size of $N = 50$.

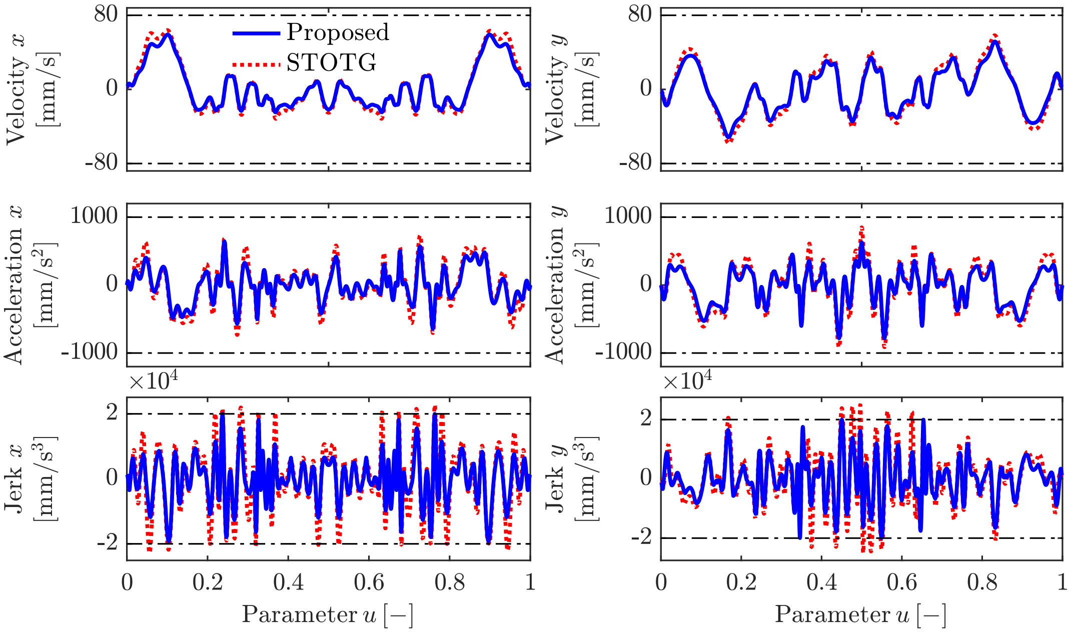


Fig. 4.8: Simulation results of velocity, acceleration, and jerk for each axis of the proposed method and STOTG with a grid size of $N = 150$.

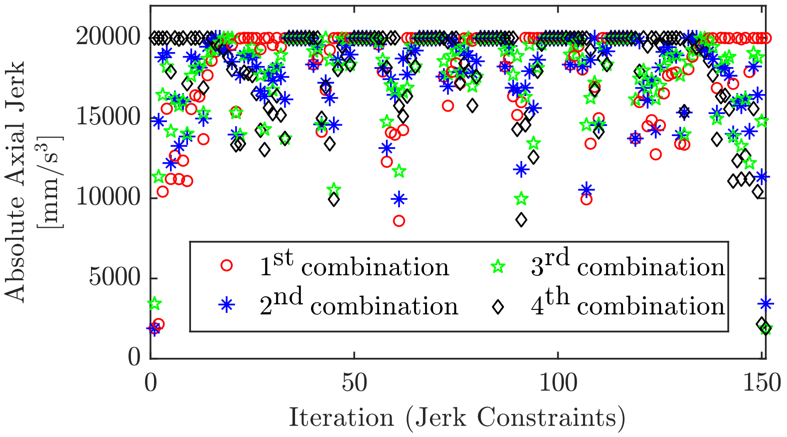


Fig. 4.9: Demonstration of the active jerk constraints in simulation by the proposed method (Jerk limit is assigned to 20000 mm/s^3).

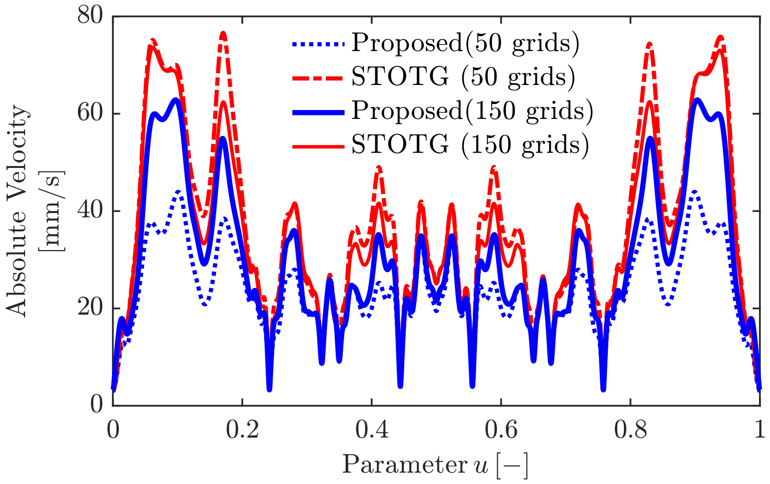


Fig. 4.10: Comparison of absolute velocities of the proposed method and STOTG with grid sizes of $N = 50$ and $N = 150$ (Simulation results).

4.5.4 Experimental verification for smoothness

In this section, the experimental results of time-optimal trajectories of the proposed and STOTG algorithms are compared for smoothness verification using the industrial bi-axial FDS (Fig. 3.14). The motion is driven by two AC servomotors in the x- and y-directions via ball-screws, which create a workspace of 170 × 200 mm. The maximum axial symmetric limits of velocity, acceleration, and jerk of the FDS are 80 mm/s, 1000 mm/s^2, and 20000 mm/s^3, respectively. The velocity and acceleration limits are determined from the specifications of motors (maximum

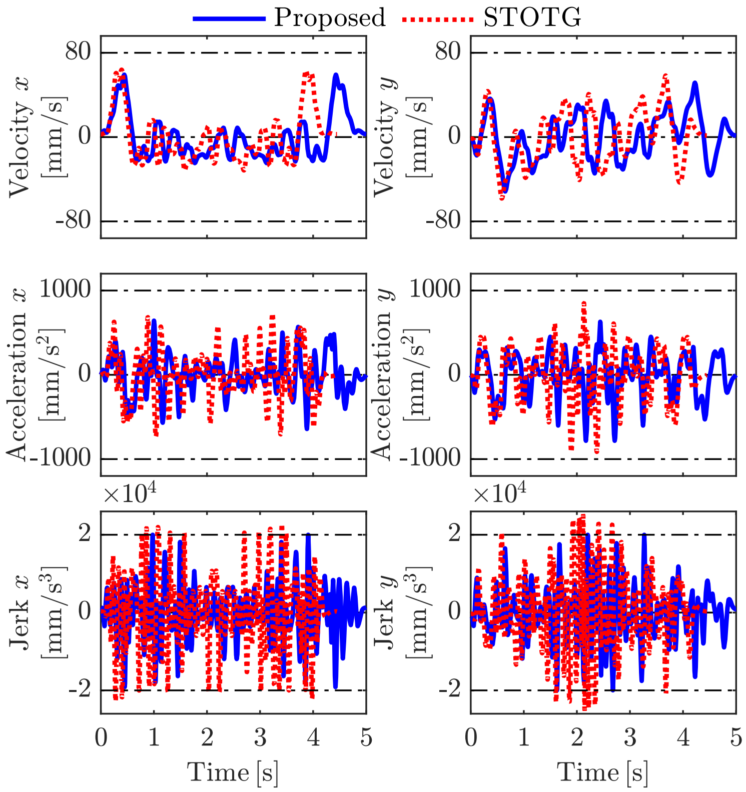


Fig. 4.11: Real-time interpolated trajectories of the proposed method and STOTG (Jerk limits are not satisfied in STOTG for the x- and y-axes).

velocity and torque) using the dynamic model of the FDS [52]. Jerk limit can be considered as a user-defined tolerance for the smoothness of the trajectory [35].

Time-optimal trajectories of the butterfly-shaped geometric path with a grid size of 150 from Section 4.5.3 were used as reference trajectories and were interpolated into real-time trajectories by the Newton–Raphson iterative method [6] (See Fig. 4.11). Observably, the STOTG violated jerk limits in the time horizon, whereas the proposed method satisfied them. The reference trajectories were discretized for each sampling time and stored as the linear interpolation data with variable feed rates. These data were fed to the FDS, controlled by a desktop computer with an Intel(R) Core i7-3770K CPU, 3.50 GHz, 8GB RAM, and a Ubuntu 15.04 64-bit operating system in a Xenomai 3.0 real-time framework. The reference positions were tracked by the same conventional PD controller with friction compensation, which used the proportional and derivative gains of 7,225 s^{-2} and 170 s^{-1}, respectively. The rotary encoders of 76.29 nm resolution were used to measure the axial positions at the sampling period of 0.2 ms.

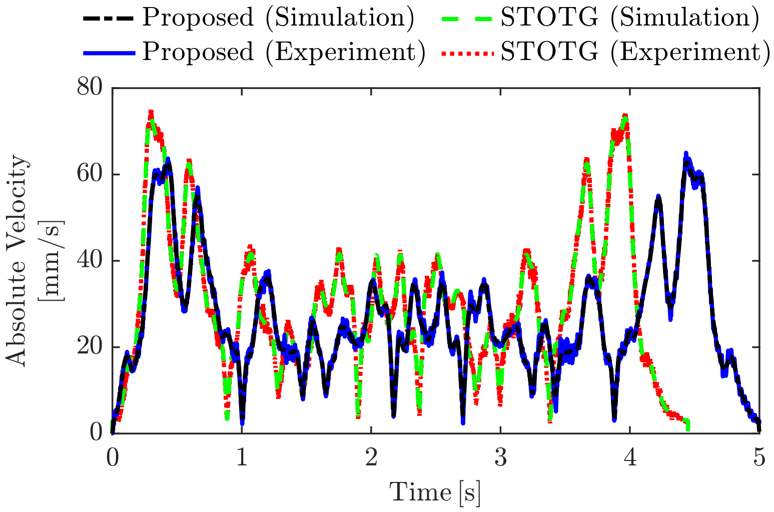


Fig. 4.12: Comparison of simulation and experimental absolute velocities of the proposed method and STOTG.

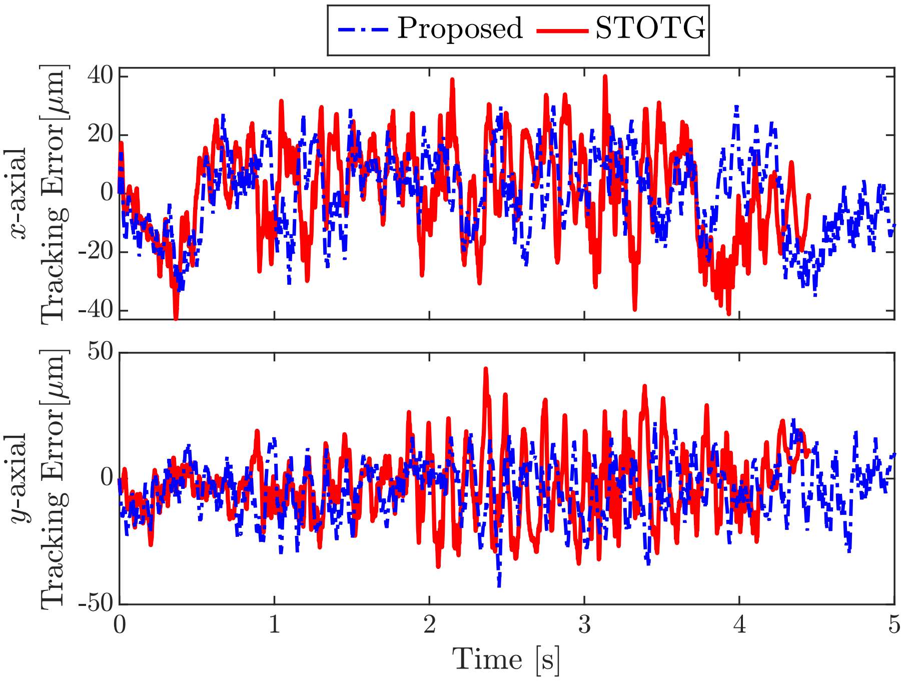


Fig. 4.13: Experimental tracking errors for x- and y-axis for the proposed method and STOTG

.

The experiments were conducted 10 times to guarantee the repeatability of the results, and the axial velocities were estimated by the numerical differentiation of the position measurements.

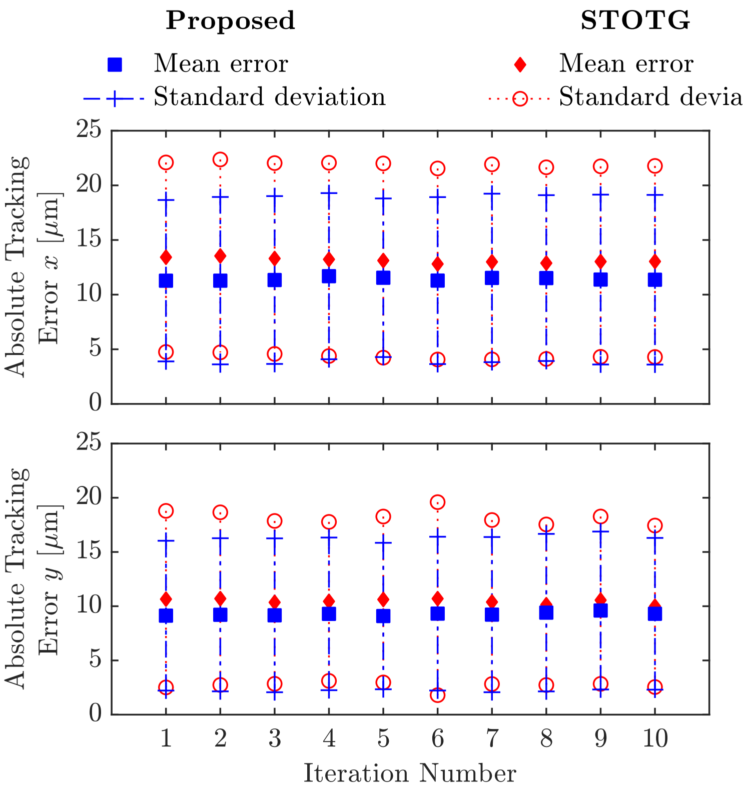


Fig. 4.14: Experimental absolute axial tracking errors of the proposed method and STOTG.

Fig. 4.12 shows the experimental absolute velocities of the proposed method and STOTG. Experimental results were similar with simulation results, and both trajectories were trackable by the PD controller. The simulation results of Section 4.5.3 showed that the STOTG violated the jerk limits, whereas the proposed method satisfied the jerk limits for the entire horizon. Therefore, the total motion time of the proposed method was approximately 11% slower than that of the STOTG. Comparing the axial acceleration and jerk values was difficult owing to the experimental noise. The trajectory with a higher jerk value induces larger vibrations in actuators affecting the tracking performance of control algorithms [41]. The tracking errors of the proposed method and STOTG are provided for all axes in Fig. 4.13. To clearly demonstrate which method has a smaller tracking error, the absolute values of the axial tracking errors were compared for both algorithms (Fig. 4.14). The mean absolute tracking errors of the proposed method in the x- and y-directions were approximately 13.07% and 11.29% smaller than those of the STOTG, respectively. Moreover, the standard deviations of the absolute tracking errors from the mean were reduced by approximately 13.43% in the x-direction and 8.99% in the

y-direction compared to those of the STOTG. These experimental results confirm that the acceleration and jerk values of the proposed method are lower than those of the STOTG and generate a smoother time-optimal trajectory that guarantees kinematic constraint satisfaction for all the entire horizon.

4.6 Summary

In this chapter, a decoupled spline-based smooth and time-optimal trajectory generation of a predefined geometric path for CNC machines was proposed in the parameter domain. The time dependency of the problem was avoided using a nonlinear transformation of variables, and the optimal times for each piecewise segment were achieved. The trajectory was parameterized using the cubic B-spline, which provided acceleration continuity for a second-order continuous geometric path and jerk continuity for a third-order continuous geometric path. A direct method was used to solve the time-OCP in the parameter horizon. We validated the accuracy of proposed kinematic constraints, time optimality, and trajectory smoothness with different problem grid sizes, kinematic constraints, and complex geometric paths. Simulations and experimental results showed the effectiveness of the proposed method over the start-of-the-art jerk-limited STOTG in [35].

Time-optimal trajectory generation of a complex predefined path is challenging due to frequent changes in derivative values. Therefore, this chapter introduces the constraint representation using spline convex hulls that limit the locally affected derivatives along the path. Compared to [35, 90, 93, 96, 97], users can select a preferable grid size of an OCP, and the proposed method provides a guaranteed solution for all kinematic constraints. Therefore, it is simple and straightforward for users. Increasing the problem grid size reduces the conservatism of kinematic constraints; therefore, the trajectories move closer to the maximum allowable kinematic limits of the machine for faster motion. A bi-axial FDS with a conventional PD controller provides a smoother trajectory with approximately 12% smaller tracking errors than the results in [35]. Thus, this trajectory generation method can be used to improve the tracking accuracy of the CNC systems, where the original controller is difficult to be changed.

Chapter 5

Coupled Approaches to Optimal Trajectory Generations with Geometric Constraints

This chapter presents coupled optimal trajectory generation approaches for CNC machine and industrial manipulator. Since decoupled approaches in Chapter 3 and 4 require the exact contour for trajectories, coupled OCPs are formulated considering the accuracy of the workpiece as a constraint.

The first part of this chapter describes a simple and effective approach for time-optimal trajectory generation of CNC machines, considering not only straight-line (G01) and circle (G02/G03) segments but also spline contour (G05) segments for a more complex profile. The cubic B-spline parameterizes the motion trajectory, and the fitting error is proposed as the orthogonal distance between the discretized via-points and the initialized curve of each G-code segment. Moreover, initial and final states, the velocity and acceleration continuity between segment trajectories, and the machine kinematic limits satisfaction are considered as equality and inequality constraints of an OCP. Thereafter, time-optimal solutions are formulated using the SQP [84].

The second part of this chapter proposes the coupled smooth trajectory generation for industrial manipulators in highly dynamic applications. With the development of robotic vision and sensor fusion technologies, research has focused on the potential of robots that can interact with humans in a safe and friendly manner in daily activities. Catching and throwing may be used in the future not only for daily activities but also in the factory instead of conventional transportation, such as an industrial robot/ belt conveyor/ gantry crane if the camera system becomes more efficient and reliable. This process provides more flexible transportation in any direction, any distance and variable speed with sufficient soft catching of an industrial product. So, this study is aimed at the industrial application as the other chapters. Planar manipulators are widely used for many purposes due to their simplicity. In this study, a three-degree of freedom planar manipulator is chosen to perform the non-prehensile object catching where the end-effector catches an object with a similar velocity to perform a soft catching in horizontal x and y-directions. Since velocity matching is required within a short time, choosing a simple and low degree of freedom manipulator for highly dynamic applications is suitable. Trajectory planning is performed in the joint space utilizing the sixth order B-spline curve, and the knot insertion is applied to match the required joint velocities for catching. Besides, minimization of the total impact force in x and y-directions to the object is proposed as an OCP considering kinematic constraints of the manipulator in the joint space.

The chapter is organized as follows: Section 5.1 briefly describes the related works for coupled OCPs; Section 5.2 presents the problem formulation of time-optimal trajectory for CNC machines; followed by the optimization results in Section 5.3. Problem formulation of smooth trajectory optimization for reduced impact catching is presented in Section 5.4, and optimization results are discussed in Section 5.5. Summary for this chapter is included in Section 5.6.

5.1 Related works

CNC machines play an important role in the modern manufacturing industry for the productivity and precision of the workpieces [1, 101]. Due to the high demands of complicated workpieces, the geometric path is created in the CAD software and converted into several G-code segments such as straight lines (G01), circles (G02/G03), and spline contours (G05). These G-code segments are usually represented by the smooth piecewise continuous functions, and the trajectory planning is considered to track the required workpiece contour considering kinematic and dynamic limitations of the machines and generate reference trajectories for CNC controllers [102, 103].

Methods for minimum time trajectory generation have been investigated in the literature to increase the productivity. Concerning the coupled approaches, Nshama *et al.* [52] proposed bi-objective trajectory generation for the linear segments considering the tolerance at cornering motions. Lu *et al.* [51] separated a given curve into several linear segments, which were represented by the S-curve trajectories with axial constraints. In [104], the high-speed cornering motion of CNC machines was presented along with the tool path contour error under axial acceleration limits. Due to the continuity, flexibility, and exact representation of complicated profiles, splines are extensively applied to parameterize the primitive motion trajectory of an OCP [25]. Gasparetto and Zanotto [7, 45] proposed a time-jerk trajectory generation of robot manipulators, satisfying kinematic limits by the convex hull property of B-splines. In [8], a smooth cornering approach for two linear segments with a required tolerance is implemented by NURBS parameterization. Mercy *et al.* [30] proposed a method for time-optimal trajectory generation for the workpiece consisting of straight-line and circle segments. However, spline contour segments were not addressed in the formulation of an OCP.

In modern manufacturing industries, the application of industrial robots become widespread due to a larger workspace, low installation cost, and higher flexibility [105]. If a robotic system can be designed to catch a flying object softly, many applications may be expected in human activities such as agriculture, livestock farming, housework and office work, because a robot or a human operator can move objects to a catching robot in the absence of any conventional transportation systems such as a conveyor or a mobile structure. In literature, there were several investigations about robotic catching which mostly focused on catching with a form or force-closure grasp [106, 107]. Concerns with non-prehensile catching, which is catching without a form or force-closure grasp, Burridge *et al.* [108] proposed three different modes: juggling, catching, and palming to catch a ball with a planar paddle. Uchiyama *et al.* [109] presented a control method of a three-degree of freedom manipulator to softly catch a falling raw egg with a small impact force in the vertical plane. Bätz *et al.* [110] predicted the ball's trajectory by the recursive least squares and achieved object catching by the balancing control method in three-dimensional space. Schill *et al.* [111] generated an initial dynamic contact between a moving object and a manipulator where the imprecise object's state is addressed.

5.2 Smooth time-optimal trajectory generation for CNC machines

5.2.1 Problem formulation

The geometric path of CNC machines consists of the three-dimensional position of the workpiece $\boldsymbol{s}(t) = [X(t), Y(t), Z(t)]^T$, where t is the motion time. The geometric path is generally divided into several G-code segments, such as straight-line segments (G01), circle segments (G02/G03), and spline contour segments (G05), and are fitted by the piecewise continuous functions to have the continuity in acceleration or jerk of the trajectory. Moreover, the velocity, acceleration, and jerk values must satisfy the upper and lower kinematic limits of CNC machines as follows:

$$\begin{aligned} \boldsymbol{v}_{\min} &\leqslant \dot{\boldsymbol{s}}(t) \leqslant \boldsymbol{v}_{\max}, \\ \boldsymbol{a}_{\min} &\leqslant \ddot{\boldsymbol{s}}(t) \leqslant \boldsymbol{a}_{\max}, \\ \boldsymbol{j}_{\min} &\leqslant \dddot{\boldsymbol{s}}(t) \leqslant \boldsymbol{j}_{\max}. \end{aligned} \tag{5.1}$$

where $\dot{\boldsymbol{s}}(t)$, $\ddot{\boldsymbol{s}}(t)$, and $\dddot{\boldsymbol{s}}(t)$ are the velocity, acceleration, and jerk of the trajectory, respectively. Moreover, the velocity and acceleration must be zero at the start and end of the trajectory, satisfying the initial and final tool positions as follows:

$$\begin{aligned} &\boldsymbol{s}(0) = \boldsymbol{q}_0,\ \boldsymbol{s}(T_{\text{tot}}) = \boldsymbol{q}_f, \\ &\dot{\boldsymbol{s}}(0) = \boldsymbol{0},\ \ \dot{\boldsymbol{s}}(T_{\text{tot}}) = \boldsymbol{0}, \\ &\ddot{\boldsymbol{s}}(0) = \boldsymbol{0},\ \ \ddot{\boldsymbol{s}}(T_{\text{tot}}) = \boldsymbol{0}. \end{aligned} \tag{5.2}$$

where $T_{\text{tot}} = \sum_{i=0}^{N} t_i$ represents the total motion time of the $N+1$ segment trajectories, $\boldsymbol{q}_0$ and $\boldsymbol{q}_f$ are the initial and final tool-tip positions, respectively. For smooth transitions between segment trajectories, the equality constraints for position, velocity, and acceleration are determined as follows:

$$\begin{aligned} \boldsymbol{s}_i(t_i) &= \boldsymbol{s}_{i+1}(0),\ \ i = \{0, 1, 2, ..., N-1\}, \\ \dot{\boldsymbol{s}}_i(t_i) &= \dot{\boldsymbol{s}}_{i+1}(0),\ \ i = \{0, 1, 2, ..., N-1\}, \\ \ddot{\boldsymbol{s}}_i(t_i) &= \ddot{\boldsymbol{s}}_{i+1}(0),\ \ i = \{0, 1, 2, ..., N-1\}. \end{aligned} \tag{5.3}$$

For the accuracy of the generated workpiece, each G-code segment is discretized into a number of m points so-called via-points, and the tool positions should satisfy the following geometric constraint at the specific via-points of the workpiece as follows:

$$\begin{aligned} &\boldsymbol{g}_{\min} \leqslant \boldsymbol{s}_i(t_{k,i}) \leqslant \boldsymbol{g}_{\max},\ k = \{0, 1, 2, ..., m\}, \\ &i = \{0, 1, 2, ..., N\}. \end{aligned} \tag{5.4}$$

where $\boldsymbol{g}_{\min}$ and $\boldsymbol{g}_{\max}$ are the minimum and maximum geometric limits, respectively. If the upper and lower kinematic limits are considered symmetric for all axes, the original OCP for minimizing the total motion time is formulated as follows:

$$\min_{\boldsymbol{s}(.),\, t_0,...,t_N} \sum_{i=0}^{N} t_i, \tag{5.5}$$

subject to (5.2)-(5.4), and

$$|\dot{\boldsymbol{s}}_i(t)| \leqslant \boldsymbol{v}_{\text{lim}},\ \forall\, t \in [0, t_i],\ i = \{0, 1, 2, ..., N\}. \tag{5.6}$$

$$|\ddot{\boldsymbol{s}}_i(t)| \leqslant \boldsymbol{a}_{\text{lim}},\ \forall\, t \in [0, t_i],\ i = \{0, 1, 2, ..., N\}. \tag{5.7}$$

$$|\dddot{\boldsymbol{s}}_i(t)| \leqslant \boldsymbol{j}_{\text{lim}},\ \forall\, t \in [0, t_i],\ i = \{0, 1, 2, ..., N\}. \tag{5.8}$$

where $\boldsymbol{v}_{\text{lim}}$, $\boldsymbol{a}_{\text{lim}}$, and $\boldsymbol{j}_{\text{lim}}$ are the symmetric velocity, acceleration, and jerk limits of the CNC machine, respectively.

5.2.2 Parameterization by cubic B-splines

This paper focuses on the time-optimal trajectory generation of CNC machines, whose contours are considered as straight line, circle, and spline segments in the x- and y-directions. In order to fit the discretized via-points for each segment $\boldsymbol{D}_i = [\boldsymbol{d}_{0,i}, \boldsymbol{d}_{1,i}, \boldsymbol{d}_{2,i}, ..., \boldsymbol{d}_{m,i}]^T$, and satisfy the velocity and acceleration constraints in (5.2) and (5.3), the cubic B-spline function with the order ($q = 4$) is used to represent the trajectory for each segment as follows:

$$\boldsymbol{s}_i(u) = \sum_{j=0}^{m+4} B_{j,q}(u)\, \boldsymbol{c}^{\text{pos}}_{j,i},\ \forall u \in [0, 1],\ i = \{0, 1, 2, ..., N\}, \tag{5.9}$$

where $\boldsymbol{s}_i(u)$ is the position with the dimensionless parameter u, $B_{j,q}(u)$ is the basis function, and $\boldsymbol{c}_{j,i}^{\text{pos}}$ represents the vector position control points of the trajectory. Thereafter, the knot vector for (5.9) can be constructed as follows:

$$\boldsymbol{u} = [\,\underbrace{0,\ ...,\ 0}_{q\text{-times}},\ u_1, u_2, ..., u_{m-1},\ \underbrace{1,\ ...,\ 1}_{q\text{-times}}\,]. \tag{5.10}$$

At the start and end of the knot vector in (5.10), the knots are q-times clamped; therefore, the first and last control points $\boldsymbol{c}_{0,i}^{\text{pos}}$ and $\boldsymbol{c}_{m+4,i}^{\text{pos}}$ are equal to the position values at $u = 0$ and $u = 1$, respectively. The inner knots are considered as uniformly distributed between the interval $[0, 1]$.

5.2.3 Initialization and determination of the fitting error

In order to fit the G-code segments described in Section 5.2.1 by the cubic B-spline, we need to determine the error between the fitted curve and the discretized via-points for each segment. In this study, the fitting error is considered as the closest distance, where the initialized curve is orthogonal to each via-point. These fitting errors will be considered as constraints of an OCP in Section 5.2.4.

Firstly, discretized via-points are assumed as control points of B-splines, repeating $(q-1)$ times at both ends, and the initialization of the curve is implemented. After that, the orthogonal distance to each via-point is computed by solving the following nonlinear equation as in [23]:

$$\begin{aligned} &(\boldsymbol{d}_{k,i} - \boldsymbol{p}_i(u)) \cdot \boldsymbol{p}_i^{(1)}(u) = 0, \\ &k = \{0, 1, 2, ..., m\},\ i = \{0, 1, 2, ..., N\}. \end{aligned} \tag{5.11}$$

where $\boldsymbol{p}_i(u)$ and $\boldsymbol{p}_i^{(1)}(u)$ are the position and first derivative of the initialized curve, respectively. Equation (5.11) is solved numerically by the Newton's iteration method and the curve parameters which satisfy the solutions are determined as $\boldsymbol{\delta}_i = [\delta_{0,i}, \delta_{1,i}, \delta_{2,i}, ..., \delta_{m,i}]^T$.

Therefore, the k^{th} fitting error for each i^{th} segment for the OCP is defined as follows:

$$\epsilon_{k,i} = ||\boldsymbol{d}_{k,i} - \boldsymbol{s}(\delta_{k,i})||_2. \tag{5.12}$$

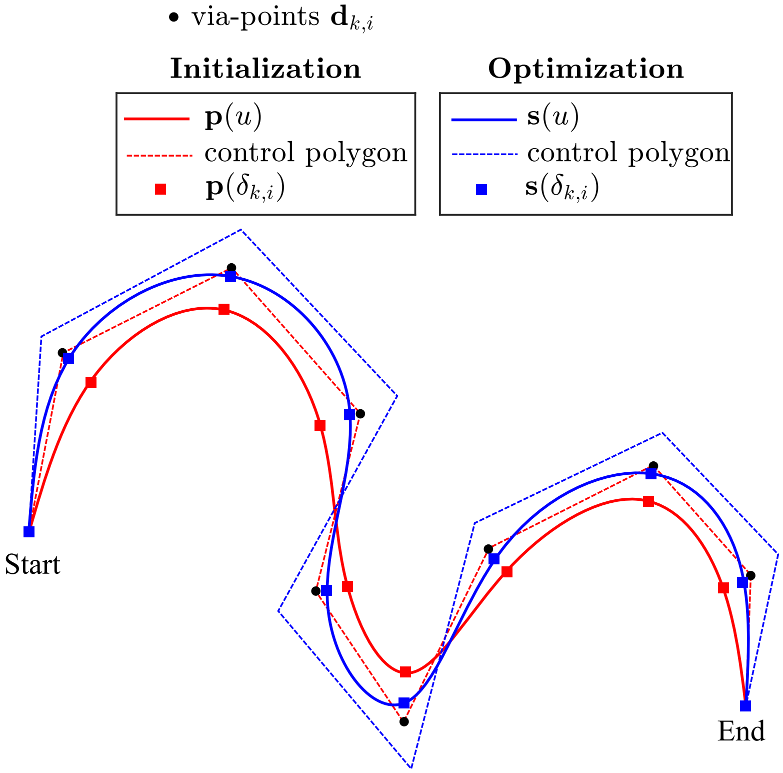


Fig. 5.1: Demonstration of discrete fitting errors for each G-code segment (Through optimization, the geometric path $\boldsymbol{s}(u)$) is closer to the discretized via-points by satisfying the fitting errors constraints).

where $||.||$ is the Euclidean norm. The fitting error $\epsilon_{k,i}$ is considered as the discrete constraints in time OCP formulation in Section 5.2.4. Demonstration for geometric path initialization and optimization considering the discrete fitting errors are shown in Fig. 5.1.

5.2.4 Spline-based time-optimal trajectory generation

The generation of time-optimal trajectories must satisfy the upper and lower kinematic limits of CNC machines for all horizons. Here, the constraints on velocity, acceleration, and jerk of the trajectory are determined using the convex hull property of B-splines [11]. A B-spline curve and its derivatives are contained within the convex hulls, determined by its control points;

therefore, we determine the respective control points for each i^{th} segment trajectory as follows:

$$\boldsymbol{c}_{j,i}^{\text{vel}} = \frac{(q-1)}{(u_{j+q} - u_{j+1})}(\boldsymbol{c}_{j+1,i}^{\text{pos}} - \boldsymbol{c}_{j,i}^{\text{pos}}),\ j = \{0, 1, ..., m+3\}. \tag{5.13}$$

$$\boldsymbol{c}_{j,i}^{\text{acc}} = \frac{(q-2)}{(u_{j+q} - u_{j+1})}(\boldsymbol{c}_{j+1,i}^{\text{vel}} - \boldsymbol{c}_{j,i}^{\text{vel}}),\ j = \{0, 1, ..., m+2\}. \tag{5.14}$$

$$\boldsymbol{c}_{j,i}^{\text{jerk}} = \frac{(q-3)}{(u_{j+q} - u_{j+1})}(\boldsymbol{c}_{j+1,i}^{\text{acc}} - \boldsymbol{c}_{j,i}^{\text{acc}}),\ j = \{0, 1, ..., m+1\}. \tag{5.15}$$

where $\boldsymbol{c}_{j,i}^{\text{vel}}$, $\boldsymbol{c}_{j,i}^{\text{acc}}$, and $\boldsymbol{c}_{j,i}^{\text{jerk}}$ denote the control points for velocity, acceleration, and jerk of each segment, respectively. For fluent transitions between segment trajectories, the following continuity constraints are adopted as in [30]:

$$\begin{aligned} \boldsymbol{s}_i(1) &= \boldsymbol{s}_{i+1}(0), \\ \dot{\boldsymbol{s}}_i(1) \cdot t_{i+1} &= \dot{\boldsymbol{s}}_{i+1}(0) \cdot t_i, \\ \ddot{\boldsymbol{s}}_i(1) \cdot t_{i+1}^2 &= \ddot{\boldsymbol{s}}_{i+1}(0) \cdot t_i^2. \end{aligned} \tag{5.16}$$

Therefore, the OCP for smooth and time-optimal trajectory generation considering fitting errors as geometric constraints is formulated as follows:

$$\min_{\boldsymbol{\phi}} \sum_{i=0}^{N} t_i, \tag{5.17}$$

where

$$\boldsymbol{\phi} = \{\boldsymbol{c}_{0,i}^{\text{pos}}, ..., \boldsymbol{c}_{m+4,i}^{\text{pos}}, t_0, ..., t_N\}. \tag{5.18}$$

subject to

$$
\begin{aligned}
&\boldsymbol{s}_0(0)=\boldsymbol{q}_0,\ \boldsymbol{s}_N(1)=\boldsymbol{q}_f,\\
&\dot{\boldsymbol{s}}_0(0)=\boldsymbol{0},\ \ \dot{\boldsymbol{s}}_N(1)=\boldsymbol{0},\\
&\ddot{\boldsymbol{s}}_0(0)=\boldsymbol{0},\ \ \ddot{\boldsymbol{s}}_N(1)=\boldsymbol{0},\\
&|\boldsymbol{c}_{j,i}^{\text{vel}}|\leqslant \boldsymbol{v}_{\text{lim}}\cdot t_i,\ j=\{0,1,...,m+3\},i=\{0,1,...,N\},\\
&|\boldsymbol{c}_{j,i}^{\text{acc}}|\leqslant \boldsymbol{a}_{\text{lim}}\cdot t_i^2,\ j=\{0,1,...,m+2\},i=\{0,1,...,N\},\\
&|\boldsymbol{c}_{j,i}^{\text{jerk}}|\leqslant \boldsymbol{j}_{\text{lim}}\cdot t_i^3,\ j=\{0,1,...,m+1\},i=\{0,1,...,N\},\\
&\epsilon_{k,i}\leqslant \epsilon_{\text{lim}},k=\{0,1,...,m\},i=\{0,1,...,N\},\\
&\boldsymbol{s}_i(1)=\boldsymbol{s}_{i+1}(0),i=\{0,1,...,N\},\\
&\dot{\boldsymbol{s}}_i(1)\cdot t_{i+1}=\dot{\boldsymbol{s}}_{i+1}(0)\cdot t_i,i=\{0,1,...,N\},\\
&\ddot{\boldsymbol{s}}_i(1)\cdot t_{i+1}^2=\ddot{\boldsymbol{s}}_{i+1}(0)\cdot t_i^2,i=\{0,1,...,N\}.
\end{aligned}
\tag{5.19}
$$

where ϵ_{lim} is the absolute limit of fitting error of the geometric path.

5.3 Optimization results and discussion

5.3.1 Calculation conditions

In order to prove the performance of the proposed method, simulation is conducted in this section. For solving the OCP described in Section 5.2.3, the rolling horizon approach [112] is used, where the solutions for the first n segments are computed, and the horizon is shifted to $n+1$ segments. The time-optimal solutions are calculated using the SQP, ("fmincon" function) in a MATLAB® environment in a laptop computer with a Windows 10 64-bit operating system with the processor of Intel(R) Core(TM) i7-7500U CPU @2.70 GHz and memory of 8 GB. The proposed method considers the fitting error constraints only at the discretized via-points of the given G-code segments.

For the application of the algorithm, we choose the letter profile TUT, representing the acronym of the Toyohashi University of Technology, which consists of 6 straight-line segments, 3 circle

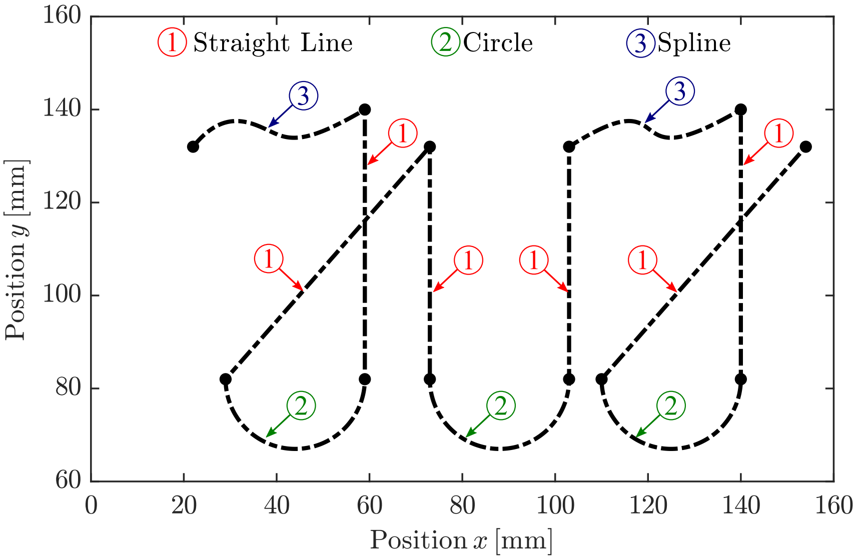


Fig. 5.2: TUT geometric profile consisting of straight-lines, circles, spline contour segments.

segments, and 2 spline contour segments as shown in Fig. 5.2. Each G-code segment is discretized into 10 via-points, and the cubic B-spline with 14 control points is used to represent each segment trajectory of the TUT profile. The symmetric velocity, acceleration, and jerk limits are set as $\boldsymbol{v}_{\text{lim}} = [80, 80]\,\text{mm/s}$, $\boldsymbol{a}_{\text{lim}} = [500, 500]\,\text{mm/s}^2$, and $\boldsymbol{j}_{\text{lim}} = [10000, 10000]\,\text{mm/s}^3$, respectively. Three segments ($n = 3$) are used for simultaneous optimization by the rolling horizon approach. The initial guess for solving the OCP is the same as the control points of initialized curve in Section 5.2.3. In this study, various fitting error constraints $\epsilon_{\text{lim}} = 0.2\,\text{mm}$ and $\epsilon_{\text{lim}} = 1\,\mu\text{m}$ are chosen for formulating the trajectory to investigate the relation between the total motion time and the accuracy of the workpiece.

5.3.2 Application results

Fig. 5.3 illustrates the x- and y-axial positions of the optimal paths with the various fitting errors $\epsilon_{\text{lim}} = 0.2\,\text{mm}$ and $\epsilon_{\text{lim}} = 1\mu\text{m}$, respectively. It is observed that both optimal paths can approximate the contour of the given TUT profile, satisfying initial and final boundary positions, including the fitting error constraint of the path for each via-point. The given geometric profile has no fluent connection between G-code segments, resulting in the discontinuity in velocity and acceleration of the trajectory. On the other hand, the proposed method satisfies the smooth

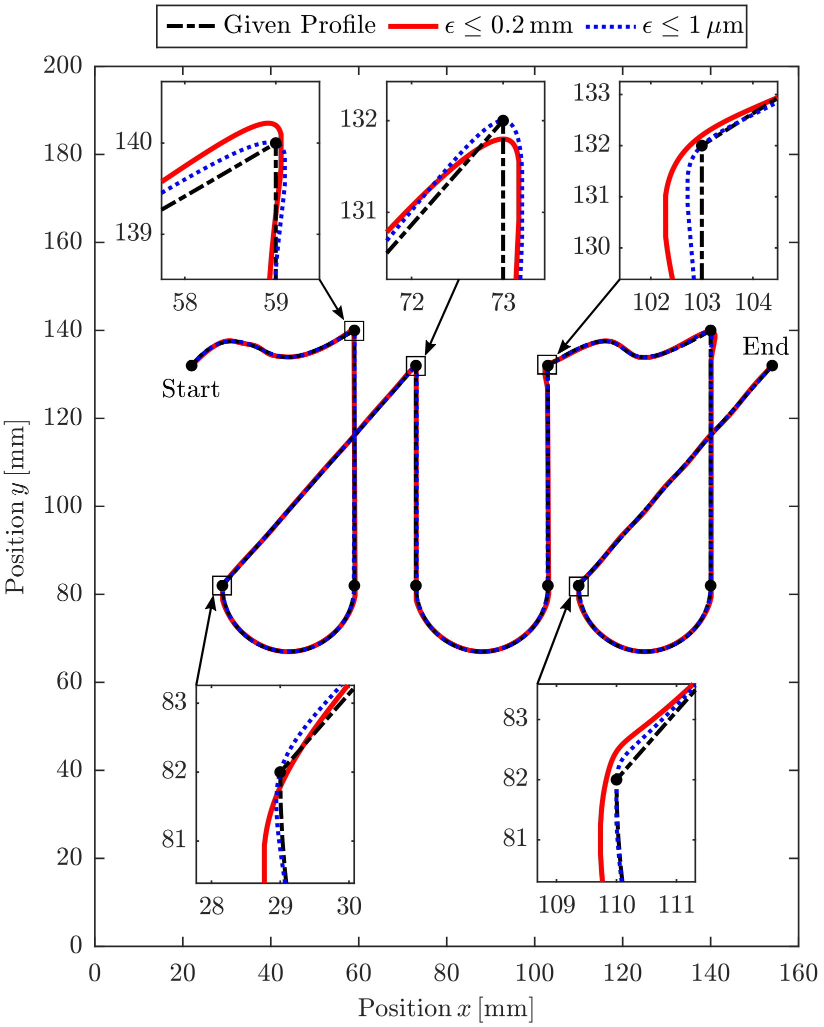
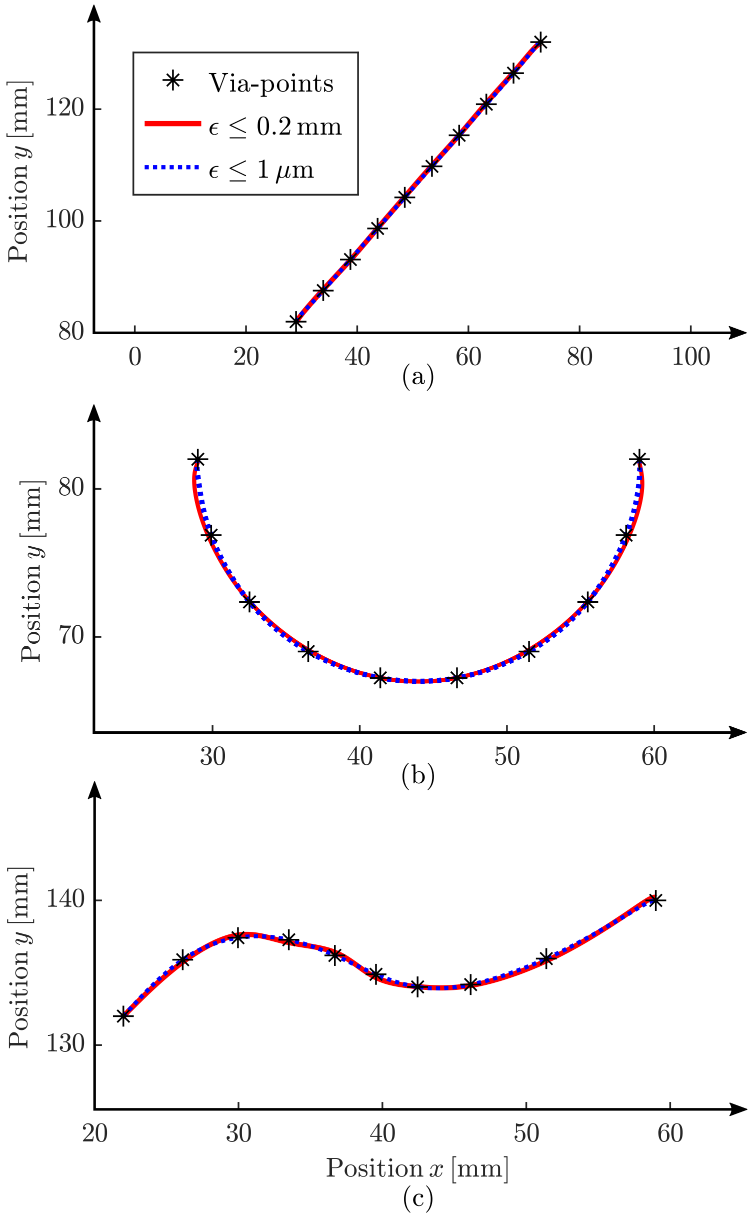


Fig. 5.3: Comparison of optimal paths and the cornering positions with different fitting errors.

transitions of the path between segment trajectories by the modification of position control points which considers not only fitting error limits but also kinematic limits of the machines. The optimal solution with the error constraint 1μm is more accurate than the solution with 0.2 mm. The resulting path passes through the corner points and all via-points within the designated accuracy, and is closer to the give G-code segments.

Fig. 5.4 shows that the proposed method can fit the straight-line (G01), circle (G02/G03), and spline contour (G05) segments compared to the via-points of the given profile while assuring the time-optimality of the generated trajectory. The time-optimal solution usually finds the shorter distance of the path with a small curvature; therefore, the algorithm can accurately fit the G01



(a)

(b)

(c)

Fig. 5.4: Fitting for (a) straight-line (G01), (b) circle (G02/G03) and (c) spline (G05) contour segments of the TUT profile with different fitting errors of 0.2 mm and 1 μm at the via-points.

segments. For G02/G03 and G05 segments, the optimal paths are slightly deviated from the via-points and in-between the via-points, depending on the limitation of the constraints of the problem. Although increasing via-points in an OCP can preserve the finer approximation of the given path, more optimization variables and constraints may increase the computational complexity of the problem.

Table 5.1: Comparison of motion times for several G-code segments (t_{G01}, $t_{\mathrm{G02/G03}}$, and t_{G05}) and the total motion time (t_{f}) according to different fitting error constraints (ϵ_{lim}).

ϵ_{lim}	t_{G01}	$t_{\mathrm{G02/G03}}$	t_{G05}	t_{f}
0.2 mm	0.802 s	0.674 s	0.953 s	8.933 s
1 μm	0.836 s	0.747 s	1.008 s	9.581 s

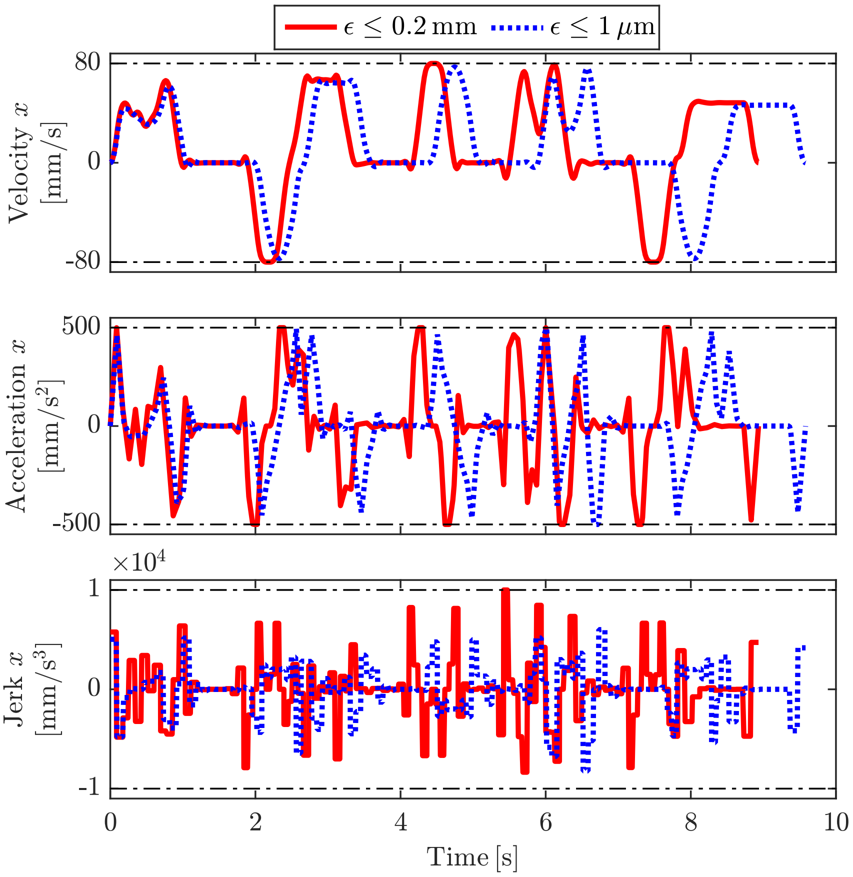


Fig. 5.5: Satisfaction of velocity, acceleration, and jerk limits on the x-axis of the TUT profile with the fitting errors of 0.2mm and 1μm.

In order to investigate the relationship between the motion time and the accuracy of the workpiece, the motion times according to different fitting error constraints are compared in Table 5.1. The optimal path with the accuracy of 0.2 mm has more flexibility dealing with the control points of B-splines than the path with 1 μm accuracy; therefore, the respective motion times of

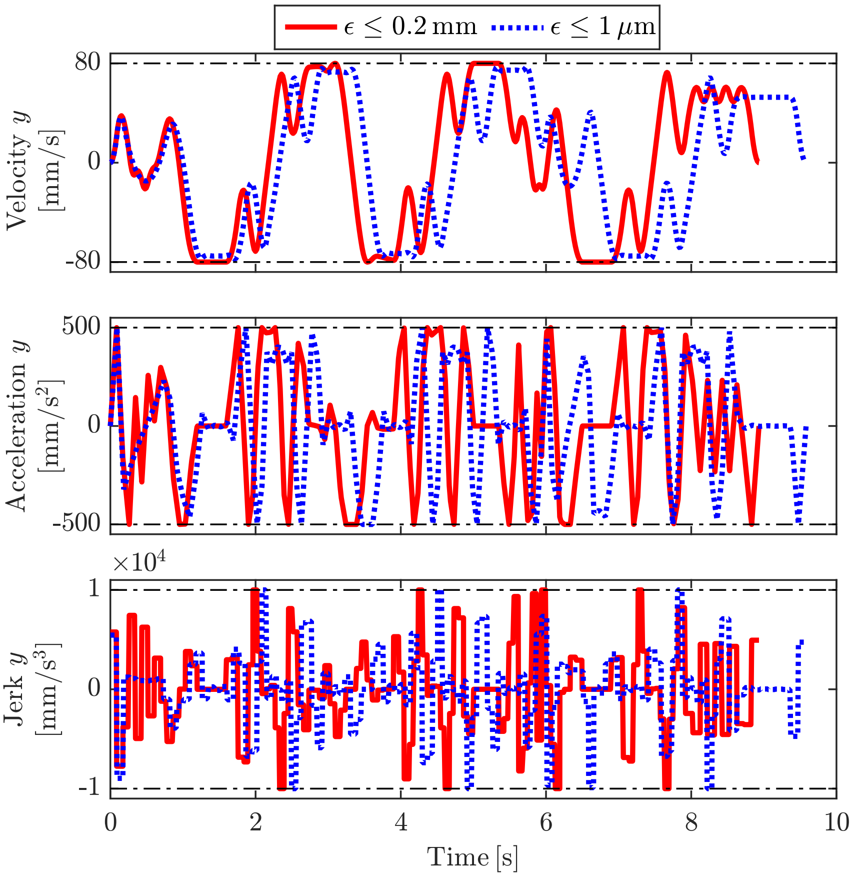


Fig. 5.6: Satisfaction of velocity, acceleration, and jerk limits on the y-axis of the TUT profile with the fitting errors of 0.2mm and 1μm.

the straight-line, circle, and spline contour segments are faster approximately 4%, 9.77%, and 5.45%, respectively. Therefore, as a result, the total motion time of 0.2 mm accuracy is 6.76% faster than the path with 1 μm accuracy under the same kinematic limits.

Figs. 5.5 and 5.6 describe the x- and y-axial velocity, acceleration, and jerk of the optimal paths with the fitting error of 0.2 mm and 1 μm, respectively. It is observed that the velocity, acceleration, and jerk values of the optimal trajectory with 0.2 mm accuracy are closer to machine kinematic limits than the trajectory with 1 μm; therefore, resulting in a faster trajectory. Both trajectories satisfy zero velocity and acceleration at the start and end of the trajectory, and the kinematic limits are satisfied for all horizons due to the convex hull property of B-splines. The continuity in axial velocities and accelerations are also satisfied along the trajectory.

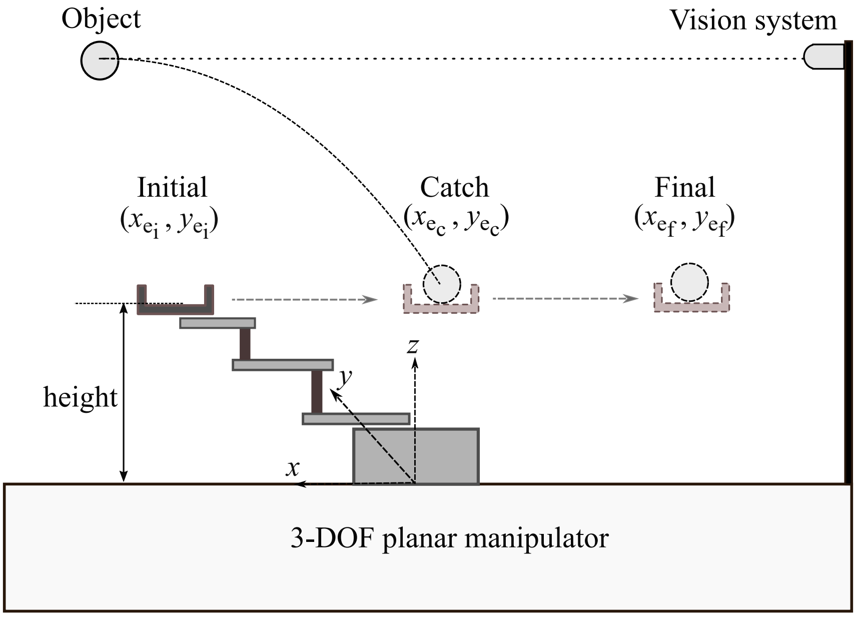


Fig. 5.7: Demonstration of reduced impact catching of a flying object.

5.4 Smooth reduced impact trajectory generation for industrial manipulators

5.4.1 Problem formulation

Throwing and catching are fundamental motions for human beings, and may be applied for advanced human and robot collaborative tasks. Since catching motion is more difficult than throwing for a robot, this study deals with reduced impact catching of a flying object by a planar manipulator. The catching position and velocity of the object are accurately estimated by the vision system, and are assumed as known parameters. An accurate prediction of the object's trajectory allows the manipulator to move from its initial position $(x_{\mathrm{e_i}}, y_{\mathrm{e_i}})$ to the catching position $(x_{\mathrm{e_c}}, y_{\mathrm{e_c}})$ within the predicted catching time t_{c}. In order to perform soft catching, the manipulator has to move to the target with the similar velocity of the object before the object falls. After the initial contact with the object, the end-effector is decelerated in the direction of $(\dot{x}_{\mathrm{e_c}}, \dot{y}_{\mathrm{e_c}})$ till the magnitude of the end-effector's velocity becomes zero at the final position $(x_{\mathrm{e_f}}, y_{\mathrm{e_f}})$ as shown in Fig. 5.7.

Therefore, the final x and y-positions of the end-effector are defined in terms of the unknown deceleration time t_d as follows:

$$\begin{aligned} x_{\mathrm{e_f}}(t_\mathrm{d}) &= x_{\mathrm{e_c}} + \tfrac{1}{2}\,\dot{x}_{\mathrm{e_c}}\,t_\mathrm{d}\,, \\ y_{\mathrm{e_f}}(t_\mathrm{d}) &= y_{\mathrm{e_c}} + \tfrac{1}{2}\,\dot{y}_{\mathrm{e_c}}\,t_\mathrm{d}\,, \end{aligned} \tag{5.20}$$

Although the catching motion has to be performed in the Cartesian space, trajectory planning is normally carried out in the joint space of the robot. Therefore, the given positions and velocities of the end-effector are transformed into corresponding joint positions and velocities by using the inverse kinematics.

Reduced impact catching of the flying object is performed during the predicted catching time t_c and deceleration time t_d. It is noticed that if t_c and t_d gets smaller, the joints of the manipulator have higher acceleration and deceleration that cause higher impact force of the end-effector to the object. The impact force in x- and y-directions according to deceleration time of the end-effector is given by

$$F_{x_\mathrm{d}} = m_\mathrm{obj}\left(\frac{\dot{x}_{\mathrm{e_f}} - \dot{x}_{\mathrm{e_c}}}{t_\mathrm{d}}\right),\ F_{y_\mathrm{d}} = m_\mathrm{obj}\left(\frac{\dot{y}_{\mathrm{e_f}} - \dot{y}_{\mathrm{e_c}}}{t_\mathrm{d}}\right), \tag{5.21}$$

where $\dot{x}_{e_f}$ and $\dot{y}_{e_f}$ denote the final velocities of the end-effector, and m_obj is the mass of the object. Here, the optimization method is considered to minimize the total impact force in horizontal x- and y-direction. According to (5.21), the impact forces are inversely proportional to the deceleration time t_d, and therefore minimization of impact forces to the object provides the optimal deceleration time and the final end-effector's position that must exist in the workspace of the manipulator. Moreover, the extrema of joint velocities and accelerations along the trajectory are bounded by their limits. The optimization problem is formulated as follows:

$$\begin{aligned} &\min_{t_\mathrm{d},\,q_j(\cdot)} \sqrt{F_{x_\mathrm{d}}^2 \;+\; F_{y_\mathrm{d}}^2}, \quad j = \{1,2,3\}, \\ &\text{subject to:} \\ &|\dot{q}_j(u)| \leqslant \dot{q}_{j,\max}, \\ &|\ddot{q}_j(u)| \leqslant \ddot{q}_{j,\max}, \\ &x_\mathrm{min} \leqslant x_{e_f} \leqslant x_\mathrm{max}, \\ &y_\mathrm{min} \leqslant y_{e_f} \leqslant y_\mathrm{max}, \end{aligned} \tag{5.22}$$

where $\dot{q}_{j,\max}$ and $\ddot{q}_{j,\max}$ are the maximum velocity and acceleration magnitudes of each joint, $[x_{\min}, x_{\max}]$ and $[y_{\min}, y_{\max}]$ are the minimum and maximum workspace limits of the end-effector in x and y-directions.

5.4.2 Trajectory paramterization by B-splines

Smooth trajectory generation is important for reduced impact catching operation. B-spline curves are widely used in trajectory planning of manipulators and CNC machines due to their continuity up to higher order derivatives of the trajectory and simple computation. Properties of B-spline curve are discussed in detail [65]. The parametric B-spline curve for the joint position with degree p, order $k = p + 1$, and number of control points $n + 1$ for the j^{th} joint is described as follows:

$$q_j(u) = \sum_{i=1}^{n+1} B_{i,k}(u)\ c_{j,i},\ \ 0 \leqslant u \leqslant 1,\ \ j = \{1, 2, 3\}, \tag{5.23}$$

where q_1, q_2 and q_3 represent the joint positions which are expressed as parameter u over the knot vector, $U = [u_1, u_2, ..., u_{n+k+1}]^T$, $c_{j,i}$ is the control point, and $B_{i,k}\,(u)$ denotes the basis function of the B-spline curve.

In order to plan a smooth catching trajectory with the continuous velocity and acceleration, a non-periodic sixth order B-spline curve is planned to interpolate the initial, catch and final joint positions, satisfy the required velocity for catching, and give the null velocity and acceleration at the initial and final boundary conditions. Since trajectory planning of B-spline curve is performed between the interval $[0, 1]$, time requirements for reduced impact catching are normalized as follows:

$$\tau_1 = 0, \quad \tau_2 = \frac{t_{\text{c}}}{t_{\text{c}} + t_{\text{d}}}, \quad \tau_3 = 1. \tag{5.24}$$

According to the distribution of assigned parameters in (5.23), the knot vector of B-spline curve is constructed. The non-periodic B-spline curve has $(p+1)$ multiple knots at both ends, and two intermediate knots are added to satisfy the required joint positions and velocities for catching. Therefore the total number of knots is $n + k + 1 = 2\,(p + 1) + 2$.

The knot vector of the sixth order B-spline curve for the catching trajectory is chosen as follows:

$$U = [\ \underbrace{\tau_1,\ ...,\ \tau_1}_{k-\text{times}},\ \frac{\tau_2}{2}, \frac{\tau_2+\tau_3}{2}\ ,\ \underbrace{\tau_3,\ ...,\ \tau_3}_{k-\text{times}}\ ]^T \tag{5.25}$$

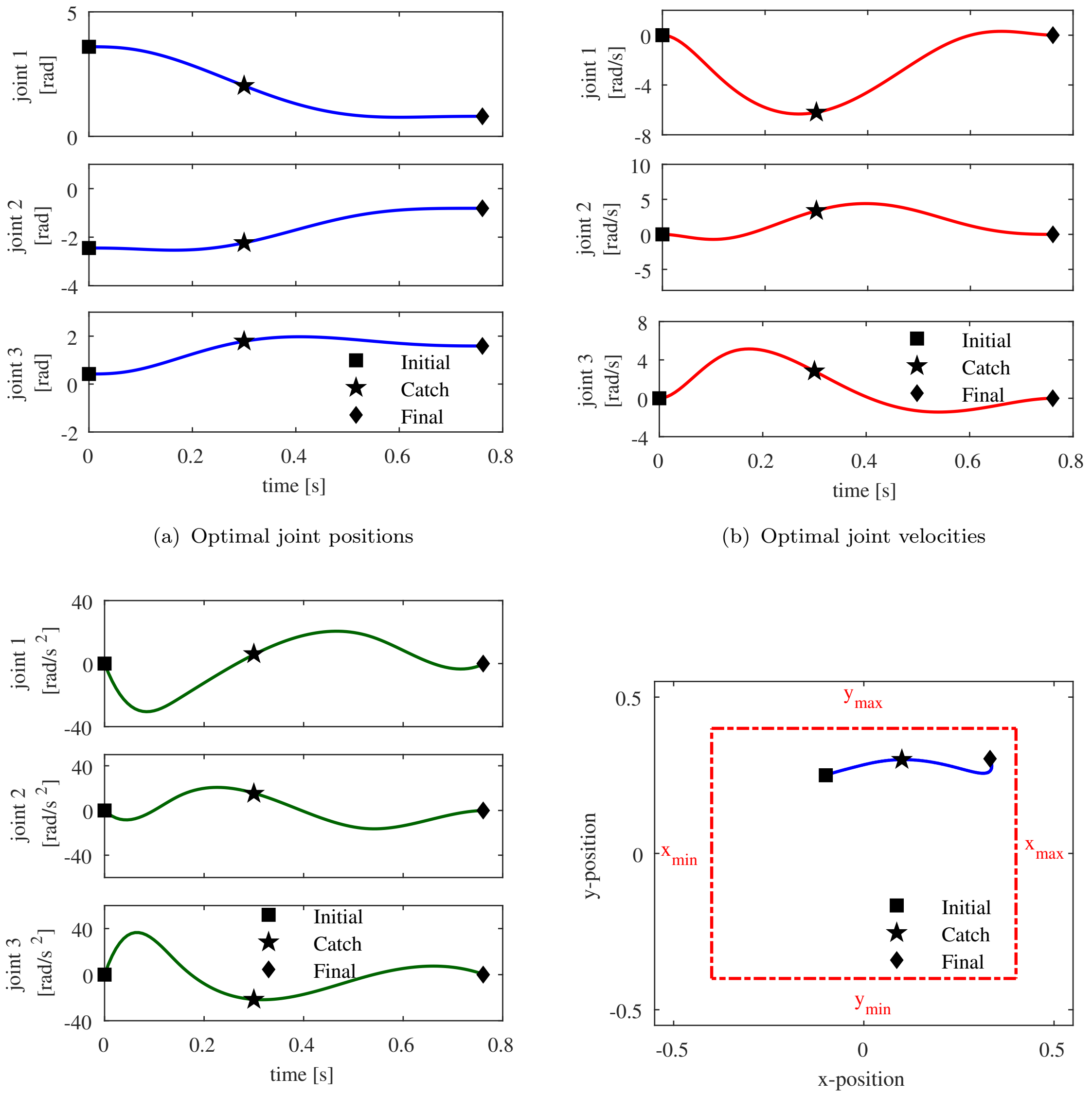
Depending on the sequence of the knot vector in (5.25), the parametric curves for the joint position, velocity and acceleration are defined. In order to find the control points of the trajectory, a set of linear equations are determined at the assigned parameters in (5.24) as follows:

$$\begin{aligned}
q_{j,\text{in}} &= c_{j,1}, \\
\dot{q}_{j,\text{in}} &= B^{(1)}_{1,k}(\tau_1)\ c_{j,1} + B^{(1)}_{2,k}(\tau_1)\ c_{j,2}, \\
\ddot{q}_{j,\text{in}} &= B^{(2)}_{1,k}(\tau_1)\ c_{j,1} + B^{(2)}_{2,k}(\tau_1)\ c_{j,2} + B^{(2)}_{3,k}(\tau_1)\ c_{j,3}, \\
q_{j,\text{c}} &= B_{2,k}(\tau_2)\ c_{j,2} + B_{3,k}(\tau_2)\ c_{j,3} + ... + B_{7,k}(\tau_2)\ c_{j,7}, \\
\dot{q}_{j,\text{c}} &= B^{(1)}_{2,k}(\tau_2)\ c_{j,2} + B^{(1)}_{3,k}(\tau_2)\ c_{j,3} + ... + B^{(1)}_{7,k}(\tau_2)\ c_{j,7}, \\
\ddot{q}_{j,\text{f}} &= B^{(2)}_{6,k}(\tau_3)\ c_{j,6} + B^{(2)}_{7,k}(\tau_3)\ c_{j,7} + B^{(2)}_{8,k}(\tau_3)\ c_{j,8}, \\
\dot{q}_{j,\text{f}} &= B^{(1)}_{7,k}(\tau_3)\ c_{j,7} + B^{(1)}_{8,k}(\tau_3)\ c_{j,8}, \\
q_{j,\text{f}} &= c_{j,8},
\end{aligned} \tag{5.26}$$

where $q_{j,\text{in}}$, $\dot{q}_{j,\text{in}}$ and $\ddot{q}_{j,\text{in}}$ denote the initial position, velocity and acceleration, $q_{j,\text{c}}$ and $\dot{q}_{j,\text{c}}$ represent the catching position and velocity, $q_{j,\text{f}}$, $\dot{q}_{j,\text{f}}$ and $\ddot{q}_{j,\text{f}}$ denote the final position, velocity and acceleration of each joint respectively.

5.5 Optimization results and discussion

To prove the performance of the optimal reduced impact catching, simulation for optimization of the trajectory is conducted with the catching time of 0.3 s and catching x and y-position of $(0.1, 0.3)$ m. The velocity limits for joint 1,2 and 3 are (18.8, 62.8, 10.5) rad/s, and for acceleration are (1600, 2142, 666) rad/s^2 respectively. The workspace limits of x and y end-effector positions are set to (± 0.4, ± 0.4) m.



(a) Optimal joint positions

(b) Optimal joint velocities

(c) Optimal joint accelerations

(d) Optimal positions in end-effector workspace

Fig. 5.8: Illustration of optimal reduced impact catching of industrial planar manipulator the joint space and the Cartesian space.

The solution of the optimization problem is obtained by using the SQP in MATLAB® environment. Within the predicted catching time, the joints are moved from the initial positions to the predicted catching positions with their target velocities. Next, they are moved to the final positions within the optimal deceleration time. The joint velocity and acceleration limits are satisfied, and the joint positions, velocities and accelerations of the optimal B-spline trajectory are shown in Figs. 5.8(a), 5.8(b) and 5.8(c), respectively. The final end-effector position is within the workspace of the manipulator as shown in Fig. 5.8(d).

5.6 Summary

The first part of this chapter presents a coupled approach for time-optimal trajectory generation of CNC machines while considering discrete fitting errors as geometric constraints of an OCP. Therefore, the proposed method enables to fit not only straight-line (G01) and circle (G02/G03) segments but also spline contour (G05) segments. Since the fitting error constraints are considered at the via-points of the G-code segments, the resulting geometric path can be slightly deviated in-between the via-points depending on the limitations of constraints, and this problem will be considered as future work. The effectiveness of the proposed method is investigated by simulations with a complicated geometric profile considering different limits on fitting errors. Smooth transitions between segment trajectories are achieved, and machine kinematic limits are satisfied for all horizons. The simulation results proved the trade-off between the time-optimality and accuracy of the workpiece by the cubic B-spline parameterization. Therefore, the proposed trajectory generation method is effective for productivity with the required accuracy of workpieces for CNC machines.

The second part of this chapter presents a coupled optimal reduced impact trajectory planning for industrial manipulators. The optimization method is provided to minimize the total impact force between the object and the end-effector in horizontal x- and y-directions. The optimal deceleration time and the final position of the end-effector, which satisfies the manipulator workspace limit, are obtained. The joint velocities and accelerations were bounded by the limits, and the resulting trajectory satisfies the required velocity at the catching position, and generates smooth and continuous trajectories.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

The modern manufacturing industries greatly rely on the CNC machines and industrial manipulators to enhance the productivity and quality of the workpieces. The goal is to track the workpiece as fast as possible while considering the kinematic and dynamic limitations of the machines and the geometric path. Previous studies addressed various approaches for optimal trajectory generations for industrial machines; however, there is a gap in the literature for determining a guaranteed technique for kinematic constraints satisfaction for all horizons, especially for decoupled OCPs. This thesis proposed a simple and efficient technique for kinematic constraint satisfaction by spline parameterization. Moreover, the method for finding the significant trade-offs from the Pareto optimal solutions was provided for the bi-objective optimization. To this end, coupled trajectory generation approaches were proposed for the practical usage of the CNC machines and industrial manipulators. A separate conclusion for each chapter is drawn as follows:

- A kinematically constrained nonlinear reparameterization approach was considered for the trade-off between total motion time and integral jerk square of the trajectories in Chapter 3. Kinematic constraints along the trajectory were satisfied, considering the spline convex hulls and the maximum geometric derivatives of the geometric path. The

Pareto front comprising the significant trade-off solutions was explored by the application of the NNC method and the D&C algorithm, where each solution was computed by the SQP. The best trade-off solution was chosen the nearest to the origin due to the high trade-off level for both objectives. The advantage of using the proposed method over linear reparameterization was discussed with simulations and experiments. Compared to the best trade-off solutions of both algorithms, the proposed method provided a 3% faster and smoother (75% lesser jerk square integral value) than the linear reparameterization. Without changing the geometric shape, the proposed method achieved the smooth starting and ending of the trajectories. Due to the lower jerk, the mean absolute tracking errors are reduced by approximately 22%.

- Chapter 4 presented the time-optimal trajectory generation for CNC machines for the complex predefined geometric paths, provided by the CAD software. This study focused on the kinematic constraint satisfaction, which was dependent on the number of grids for constraints by previous studies. The OCP was formulated in the parameter domain in order to avoid the time-dependency. Our previous kinematic constraints in Chapter 3 were highly conservative due to the maximum geometric derivatives of the path. For this reason, the kinematic constraints were proposed based on the spline convex hulls that limited the locally affected geometric derivatives; therefore, it applied to the complex geometric paths, whose derivatives were frequently changed within the parameter horizon. The comparison was made with the STOTG in the parameter domain by applying various geometric paths and problem grids. The simulation results showed that the proposed method was more robust to the problem grids and guaranteed the kinematic constraints for all horizons. The experimental results validated that the proposed method provides a smoother trajectory, reducing the mean absolute tracking errors by approximately 12%.

- Chapter 5 discussed two coupled approaches for optimal trajectory generation. The first approach was the contour reshaping of the workpieces, consisting of the straight-line (G01), circle (G02/G03), and spline contour (G05) segments while minimizing the total motion time by considering the kinematic and geometric constraints along the path. The geometric constraints were proposed as the closest distance between the discretized via-points of each G-code segment and the initialized B-spline curve. The continuity constraints for velocity and acceleration achieved smooth transitions between segment trajectories. The rolling horizon approach was used to optimize multiple segment trajectories simultaneously. The optimization results showed that trajectory with tight geometric constraints better approximated the geometric shape; however, it suffered from slower motion times. Therefore, the trade-off between time-optimality and accuracy of the workpiece was confirmed. The second approach was the smooth trajectory generation for reduced impact motion for industrial manipulators. After catching a flying object with a similar velocity, the OCP was formulated to reduce the impact force, which was dependent on the deceleration time and final position of the motion. The optimization results showed that the required velocities for the catching position are satisfied, and the joint velocities and accelerations are bounded by the maximum limits. The optimal deceleration time decided the ending position of reduced impact motion with zero velocities and acceleration, satisfying the workspace limits.

6.2 Future works

The proposed methods can be further improved or extended by considering the following facts:

- All control parameterizations by B-splines used in this thesis consider the uniform distribution of the knot points for simplicity. If the knots are considered as optimization variables, the optimal motion times for each piecewise segment can be computed independently. Therefore, a faster total motion time may be achieved.
- In Chapter 3, the conservatism of kinematic constraints is reduced by increasing the problem grids; however, as a trade-off, the computation time increases to solve the problem. The nonlinear optimization solver SQP may fail to find a feasible solution when the problem grids are very large (e.g., $N = 10000$). Therefore, suitable grids within $50 \sim 200$ are recommended depending on the geometric path. In contrast, the complexity of the large grids OCP can be reduced by dividing the optimization horizon into several parts and solving by the simultaneous optimization techniques, such as the rolling horizon approach as in Section 5.2. This procedure can be left as future work.

- The single objective OCPs proposed in Chapters 4 and 5 can be extended to the multi-objective OCPs considering the contradictory objectives of the motion time such as energy consumption, the accuracy of the workpiece, and smoothness of the trajectories.

- The generated optimal trajectory represents the motion of the tool tip position along the geometric paths. Therefore, this studies can be applied to the optimal trajectory generation of the tool path for the three-axis or five-axis CNC machines. If the tool orientation is represented by the rotational axis trajectory, the proposed method can be applied to five-axis machines as well. This application can be left as future work.

- This thesis focuses on the spline-based optimal reference trajectory generation considering kinematic constraints and the workpiece accuracy (in coupled approaches). However, the machining constraints such as spindle speed, cutting width and depth, cutting force, and the surface roughness [113, 114] are not considered in the OCP formulation. The reference optimal trajectory can be applied for machining as in [30], but the experimental results may deviate from the simulation (optimization results). To avoid this drawback, the author recommends considering the abovementioned machining constraints in the OCP formulation as a future work.

- Considering the computational complexity, this study does not consider the dynamic constraints in the OCP. Tracking errors are reduced by limiting or minimizing the jerk related objectives. On the other hand, tracking error constraints by the typical controllers can be included in the OCP formulation as in [39, 40].

- The geometric constraints are considered at the via-points of the G-codes segments in Section 5.2. According to the bang-bang behavior of the kinematic constraints in time-optimal trajectories, the resulting geometric path may violate the fitting error constraints in-between the discretized via-points. This investigation is left as future work.

- The optimal reference trajectories are generated for the reduced impact catching of a flying object in Section 5.4. However, the successful catching mainly depends on the accurate estimation of the object's trajectory by the vision system, and the movement of the manipulator within the designated catching time. Since the time for the catching operation is very small (approximately 0.3 s in our system), it is difficult to optimize the trajectory online. Therefore, the author recommends to conduct trajectory optimization off-line, and perform the real-time modification of the trajectory as in [115], because it saves much computational time without recomputing the entire trajectory. This is left as a future work.

Bibliography

[1] T. V. Vardhan and B. S. Babu, “Multiaxis cnc programming and machining,” in *Modern Manufacturing Processes*. UK: Woodhead Publishing, pp. 167–175, 2020.

[2] M. Yang, E. Yang, R. C. Zante, M. Post, and X. Liu, “Collaborative mobile industrial manipulator: a review of system architecture and applications,” in *25th International Conference on Automation and Computing*, pp. 1–6, 2019.

[3] Y. Chen and F. Dong, “Robot machining: recent development and future research issues,” *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9, pp. 1489–1497, 2013.

[4] Y. Altintas and A. Ber, *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*. USA: Cambridge University Press, 2000.

[5] Z. Shiller and H.-H. Lu, “Computation of path constrained time optimal motions with dynamic singularities,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 114, no. 1, pp. 34–40, 1992.

[6] S. D. Timar, R. T. Farouki, T. S. Smith, and C. L. Boyadjieff, “Algorithms for time–optimal control of cnc machines along curved tool paths,” *Robotics and Computer-Integrated Manufacturing*, vol. 21, no. 1, pp. 37–53, 2005.

[7] A. Gasparetto and V. Zanotto, “A new method for smooth trajectory planning of robot manipulators,” *Mechanism and Machine Theory*, vol. 42, no. 4, pp. 455–471, 2007.

[8] M. Duan and C. Okwudire, “Minimum-time cornering for cnc machines using an optimal control method with nurbs parameterization,” *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 5, pp. 1405–1418, 2016.

[9] S. Yang, A. H. Ghasemi, X. Lu, and C. E. Okwudire, “Pre-compensation of servo contour errors using a model predictive control framework,” *International Journal of Machine Tools and Manufacture*, vol. 98, pp. 50–60, 2015.

[10] A. A. E. K. Mohammad, N. Uchiyama, and S. Sano, “Energy saving in feed drive systems using sliding-mode-based contouring control with a nonlinear sliding surface,” *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 572–579, 2014.

[11] L. Piegl and W. Tiller, *The NURBS Book*. Springer, 1995.

[12] M. M.-T. Heng, “Smooth and time-optimal trajectory generation for high speed machine tools,” Master’s Thesis, University of Waterloo, 2008.

[13] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. MIT Press, 2011.

[14] C.-W. Cheng and M.-C. Tsai, “Real-time variable feed rate nurbs curve interpolator for cnc machining,” *The International Journal of Advanced Manufacturing Technology*, vol. 23, no. 11, pp. 865–873, 2004.

[15] B. Sencer, “Trajectory generation for cnc machine tools,” *Journal of the Japan Society for Precision Engineering*, vol. 79, no. 7, pp. 631–638, 2013.

[16] B. A. Barsky and T. D. DeRose, *Geometric Continuity of Parametric Curves*. University of California at Berkeley, 1984.

[17] B. A. Barsky and T. D. DeRose, “Geometric continuity of parametric curves: three equivalent characterizations,” *IEEE Computer Graphics and Applications*, vol. 9, no. 6, pp. 60–69, 1989.

[18] S.-R. Chang and U.-Y. Huh, “G2 continuity smooth path planning using cubic polynomial interpolation with membership function,” *Journal of Electrical Engineering and Technology*, vol. 10, no. 2, pp. 676–687, 2015.

[19] Y. Chen, Y. Cai, J. Zheng, and D. Thalmann, “Accurate and efficient approximation of clothoids using bézier curves for path planning,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1242–1247, 2017.

[20] J. M. Langeron, E. Duc, C. Lartigue, and P. Bourdet, "A new format for 5-axis tool path computation, using bspline curves," *Computer-Aided Design*, vol. 36, no. 12, pp. 1219–1229, 2004.

[21] W. Wang, H. Pottmann, and Y. Liu, "Fitting b-spline curves to point clouds by curvature-based squared distance minimization," *ACM Transactions on Graphics*, vol. 25, no. 2, pp. 214–238, 2006.

[22] H. Pottmann, S. Leopoldseder, and M. Hofer, "Approximation with active b-spline curves and surfaces," in *10th Pacific Conference on Computer Graphics and Applications*, pp. 8–25, 2002.

[23] S.-I. Gofuku, S. Tamura, and T. Maekawa, "Point-tangent/point-normal b-spline curve interpolation by geometric algorithms," *Computer-Aided Design*, vol. 41, no. 6, pp. 412–422, 2009.

[24] S. Okaniwa, A. Nasri, H. Lin, A. Abbas, Y. Kineri, and T. Maekawa, "Uniform b-spline curve interpolation with prescribed tangent and curvature vectors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1474–1487, 2011.

[25] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Springer, 2008.

[26] Q. G. Zhang and R. B. Greenway, "Development and implementation of a nurbs curve motion interpolator," *Robotics and Computer-Integrated Manufacturing*, vol. 14, no. 1, pp. 27–36, 1998.

[27] W. Lei, M. Sung, L. Lin, and J. Huang, "Fast real-time nurbs path interpolation for cnc machine tools," *International Journal of Machine Tools and Manufacture*, vol. 47, no. 10, pp. 1530–1541, 2007.

[28] M. Heng and K. Erkorkmaz, "Design of a nurbs interpolator with minimal feed fluctuation and continuous feed modulation capability," *International Journal of Machine Tools and Manufacture*, vol. 50, no. 3, pp. 281–293, 2010.

[29] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, and W. Burgard, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[30] T. Mercy, N. Jacquod, R. Herzog, and G. Pipeleers, "Spline-based trajectory generation for cnc machines," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, pp. 6098–6107, 2018.

[31] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[32] E. Jamhour and P. André, "Planning smooth trajectories along parametric paths," *Mathematics and Computers in Simulation*, vol. 41, no. 5-6, pp. 615–626, 1996.

[33] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE Journal on Robotics and Automation*, vol. 3, no. 2, pp. 115–123, 1987.

[34] Z. Shiller, "Time-energy optimal control of articulated systems with geometric path constraints," *Journal of Dynamic Systems, Measurement, and Control*, vol. 118, no. 1, pp. 139–143, 1996.

[35] Q. Zhang, S. Li, and J. Guo, "Smooth time-optimal tool trajectory generation for cnc manufacturing systems," *Journal of Manufacturing Systems*, vol. 31, no. 3, pp. 280–287, 2012.

[36] J.-W. Ma, S. Gao, H.-T. Yan, Q. Lv, and G.-Q. Hu, "A new approach to time-optimal trajectory planning with torque and jerk limits for robot," *Robotics and Autonomous Systems*, vol. 140, 2021.

[37] H. Liu, X. Lai, and W. Wu, "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 2, pp. 309–317, 2013.

[38] D. Constantinescu and E. A. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *Journal of Robotic Systems*, vol. 17, no. 5, pp. 233–249, 2000.

[39] K. Zhang, J.-X. Guo, and X.-S. Gao, "Cubic spline trajectory generation with axis jerk and tracking error constraints," *International Journal of Precision Engineering and Manufacturing*, vol. 14, no. 7, pp. 1141–1146, 2013.

[40] Q. Zhang, S. Li, and J. Guo, “Minimum time trajectory optimization of cnc machining with tracking error constraints,” *Abstract and Applied Analysis*, vol. 2014, 2014.

[41] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, *Path Planning and Trajectory Planning Algorithms: A General Overview.* Springer, pp. 3–27, 2015.

[42] K. J. Kyriakopoulos and G. N. Saridis, “Minimum jerk path generation,” in *IEEE International Conference on Robotics and Automation*, pp. 364–369, 1988.

[43] A. Piazzi and A. Visioli, “An interval algorithm for minimum-jerk trajectory planning of robot manipulators,” in *36th IEEE Conference on Decision and Control*, vol. 2, pp. 1924–1927, 1997.

[44] A. Piazzi and A. Visioli, “Global minimum-jerk trajectory planning of robot manipulators,” *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 140–149, 2000.

[45] A. Gasparetto and V. Zanotto, “A technique for time-jerk optimal planning of robot trajectories,” *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 415–426, 2008.

[46] A. Gasparetto, A. Lanzutti, R. Vidoni, and V. Zanotto, “Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning,” *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 164–181, 2012.

[47] F. Sellmann, “Exploitation of tolerances and quasi-redundancy for set point generation,” Ph.D. Dissertation, ETH Zurich, 2014.

[48] Y. Yan, L. Zhang, and K. Zhang, “Corner smoothing transition algorithm for five-axis linear tool path,” *Procedia CIRP*, vol. 56, pp. 604–609, 2016.

[49] B. Sencer, K. Ishizaki, and E. Shamoto, “A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of nc systems along linear tool paths,” *The International Journal of Advanced Manufacturing Technology*, vol. 76, no. 9, pp. 1977–1992, 2015.

[50] S. Tajima and B. Sencer, “Kinematic corner smoothing for high speed machine tools,” *International Journal of Machine Tools and Manufacture*, vol. 108, pp. 27–43, 2016.

[51] T.-C. Lu, S.-L. Chen, and E. C.-Y. Yang, “Near time-optimal s-curve velocity planning for multiple line segments under axis constraints,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9582–9592, 2018.

[52] E. W. Nshama and N. Uchiyama, “Pareto optimization of cycle time and motion accuracy in trajectory planning for industrial feed drive systems,” *IEEE Access*, vol. 9, pp. 114 104–114 119, 2021.

[53] D. E. Kirk, *Optimal Control Theory: An Introduction*. Courier Corporation, 2012.

[54] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, “Fast direct multiple shooting algorithms for optimal robot control,” in *Fast Motions in Biomechanics and Robotics*. Springer, pp. 65–93, 2006.

[55] P. Drag, K. Styczeń, M. Kwiatkowska, and A. Szczurek, “A review on the direct and indirect methods for solving optimal control problems with differential-algebraic constraints,” in *Recent Advances in Computational Optimization*. Springer, pp. 91–105, 2016.

[56] M. Korayem, V. Azimirad, A. Nikoobin, and Z. Boroujeni, “Maximum load-carrying capacity of autonomous mobile manipulator in an environment with obstacle considering tip over stability,” *The International Journal of Advanced Manufacturing Technology*, vol. 46, no. 5-8, pp. 811–829, 2010.

[57] M. H. Korayem, M. Nazemizadeh, and H. R. Nohooji, “Smooth jerk-bounded optimal path planning of tricycle wheeled mobile manipulators in the presence of environmental obstacles,” *International Journal of Advanced Robotic Systems*, vol. 9, no. 4, 2012.

[58] M. H. Korayem, H. R. Nohooji, and A. Nikoobin, “Path planning of mobile elastic robotic arms by indirect approach of optimal control,” *International Journal of Advanced Robotic Systems*, vol. 8, no. 1, pp. 10–20, 2011.

[59] M. H. Korayem and H. Ghariblu, “Maximum allowable load on wheeled mobile manipulators imposing redundancy constraints,” *Robotics and Autonomous Systems*, vol. 44, no. 2, pp. 151–159, 2003.

[60] M. Posa, C. Cantu, and R. Tedrake, “A direct method for trajectory optimization of rigid bodies through contact,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[61] Q. Gong, W. Kang, N. S. Bedrossian, F. Fahroo, P. Sekhavat, and K. Bollino, "Pseudospectral optimal control for military and industrial applications," in *46th IEEE Conference on Decision and Control*, pp. 4128–4142, 2007.

[62] M. P. Kelly, "Transcription methods for trajectory optimization: a beginners tutorial," *arXiv preprint arXiv:1707.00284*, 2017.

[63] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.

[64] C. De Boor, *A Practical Guide to Splines*. Springer-Verlag, vol. 27, 1978.

[65] D. F. Rogers, *An Introduction to NURBS: with Historical Perspective*. Academic Press, 2001.

[66] Q.-X. Huang, S.-M. Hu, and R. R. Martin, "Fast degree elevation and knot insertion for b-spline curves," *Computer Aided Geometric Design*, vol. 22, no. 2, pp. 183–197, 2005.

[67] T. Mercy, W. Van Loock, and G. Pipeleers, "Real-time motion planning in the presence of moving obstacles," in *2016 European Control Conference*, pp. 1586–1591, 2016.

[68] S. F. Hosseini, B. Moetakef-Imani, S. Hadidi-Moud, and B. Hassani, "The effect of parameterization on isogeometric analysis of free-form curved beams," *Acta Mechanica*, vol. 227, no. 7, pp. 1983–1998, 2016.

[69] M. Randrzanarivony and G. Brunnen, "Approximation by nurb curves with free knots." in *Vision, Modeling, and Visualization*, pp. 195–201, 2002.

[70] B. Imani and S. A. Hashemian, "Nurbs-based profile reconstruction using constrained fitting techniques," *Journal of Mechanics*, vol. 28, no. 3, pp. 407–412, 2012.

[71] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010.

[72] O. Grodzevich and O. Romanko, "Normalization and other topics in multi-objective optimization," in *Fields-MITACS Industrial Problems Workshop*, pp. 89–101, 2006.

[73] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems," *Structural Optimization*, vol. 14, no. 1, pp. 63–69, 1997.

[74] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[75] A. Messac, A. Ismail-Yahaya, and C. A. Mattson, "The normalized normal constraint method for generating the pareto frontier," *Structural and Multidisciplinary Optimization*, vol. 25, no. 2, pp. 86–98, 2003.

[76] C. A. Mattson, A. A. Mullur, and A. Messac, "Smart pareto filter: obtaining a minimal representation of multiobjective design space," *Engineering Optimization*, vol. 36, no. 6, pp. 721–740, 2004.

[77] E. Antipova, C. Pozo, G. Guillén-Gosálbez, D. Boer, L. F. Cabeza, and L. Jiménez, "On the use of filters to facilitate the post-optimal analysis of the pareto solutions in multi-objective optimization," *Computers & Chemical Engineering*, vol. 74, pp. 48–58, 2015.

[78] I. Hashem, D. Telen, P. Nimmegeers, F. Logist, and J. Van Impe, "A novel algorithm for fast representation of a pareto front with adaptive resolution: Application to multi-objective optimization of a chemical reactor," *Computers & Chemical Engineering*, vol. 106, pp. 544–558, 2017.

[79] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[80] A. Hashemian, S. F. Hosseini, and S. N. Nabavi, "Kinematically smoothing trajectories by nurbs reparameterization–an innovative approach," *Advanced Robotics*, vol. 31, no. 23-24, pp. 1296–1312, 2017.

[81] S. S. De, S. Dehuri, and S.-B. Cho, "Divide-and-conquer approach for revealing the non-dominated solutions in multi-objective optimization problem," in *2017 International Conference on Information Technology*, pp. 143–151, 2017.

[82] E. W. Nshama, M. R. Msukwa, and N. Uchiyama, “A trade-off between energy saving and cycle time reduction by pareto optimal corner smoothing in industrial feed drive systems,” *IEEE Access*, vol. 9, pp. 23 579–23 594, 2021.

[83] J. Sanchis, M. Martinez, X. Blasco, and J. Salcedo, “A new perspective on multiobjective optimization by enhanced normalized normal constraint method,” *Structural and Multidisciplinary Optimization*, vol. 36, no. 5, pp. 537–546, 2008.

[84] P. E. Gill and E. Wong, “Sequential quadratic programming methods,” in *Mixed Integer Nonlinear Programming*. Springer, pp. 147–224, 2012.

[85] W. Aribowo and K. Terashima, “Cubic spline trajectory planning and vibration suppression of semiconductor wafer transfer robot arm,” *International Journal of Automation Technology*, vol. 8, no. 2, pp. 265–274, 2014.

[86] R. V. Fleisig and A. D. Spence, “A constant feed and reduced angular acceleration interpolation algorithm for multi-axis machining,” *Computer-Aided Design*, vol. 33, no. 1, pp. 1–15, 2001.

[87] K. Erkorkmaz and Y. Altintas, “High speed cnc system design. part i: jerk limited trajectory generation and quintic spline interpolation,” *International Journal of machine tools and manufacture*, vol. 41, no. 9, pp. 1323–1345, 2001.

[88] I. Giagkiozis and P. J. Fleming, “Methods for multi-objective optimization: An analysis,” *Information Sciences*, vol. 293, pp. 338–350, 2015.

[89] M. S. Selig, *UIUC Applied Aerodynamics Group*, https://m-selig.ae.illinois.edu/index.html [Accessed: January 2021].

[90] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, “Time-optimal path tracking for robots: a convex optimization approach,” *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[91] P. T. Boggs and J. W. Tolle, “Sequential quadratic programming for large-scale nonlinear optimization,” *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 123–137, 2000.

[92] M. S. Paing, E. W. Nshama, and N. Uchiyama, “A kinematically constrained reparameterization approach to optimal time and jerk motion of industrial machines,” *IEEE Access*, vol. 9, pp. 97 843–97 854, 2021.

[93] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, “Time-energy optimal path tracking for robots: a numerically efficient optimization approach,” in *10th IEEE International Workshop on Advanced Motion Control*, pp. 727–732, 2008.

[94] K. R. Simba, B. D. Bui, M. R. Msukwa, and N. Uchiyama, “Robust iterative learning contouring controller with disturbance observer for machine tool feed drives,” *ISA Transactions*, vol. 75, pp. 207–215, 2018.

[95] A. Farrage and N. Uchiyama, “Improvement of motion accuracy and energy consumption of a mechanical feed drive system using a fourier series-based nonlinear friction model,” *The International Journal of Advanced Manufacturing Technology*, vol. 99, no. 5, pp. 1203–1214, 2018.

[96] W. Fan, X.-S. Gao, C.-H. Lee, K. Zhang, and Q. Zhang, “Time-optimal interpolation for five-axis cnc machining along parametric tool path based on linear programming,” *The International Journal of Advanced Manufacturing Technology*, vol. 69, no. 5, pp. 1373–1388, 2013.

[97] J. Dong, P. M. Ferreira, and J. A. Stori, “Feed-rate optimization with jerk constraints for generating minimum-time trajectories,” *International Journal of Machine Tools and Manufacture*, vol. 47, no. 12-13, pp. 1941–1955, 2007.

[98] D. E. Kirk, *Optimal Control Theory: An Introduction.* Courier Corporation, 2012.

[99] T. W. Chen and V. S. Vassiliadis, “Inequality path constraints in optimal control: a finite iteration ε-convergent scheme based on pointwise discretization,” *Journal of Process Control*, vol. 15, no. 3, pp. 353–362, 2005.

[100] W. Van Loock, G. Pipeleers, and J. Swevers, “B-spline parameterized optimal motion trajectories for robotic systems with guaranteed constraint satisfaction,” *Mechanical Sciences*, vol. 6, no. 2, pp. 163–171, 2015.

[101] Y. Altintas, A. Verl, C. Brecher, L. Uriarte, and G. Pritschow, “Machine tool feed drives,” *CIRP Annals*, vol. 60, no. 2, pp. 779–796, 2011.

[102] B. D. Bui, N. Uchiyama, and K. R. Simba, “Contouring control for three-axis machine tools based on nonlinear friction compensation for lead screws,” *International Journal of Machine Tools and Manufacture*, vol. 108, pp. 95–105, 2016.

[103] A. Farrage and N. Uchiyama, “Improvement of motion accuracy and energy consumption for industrial feed drive systems using adaptive sliding mode control,” *ISA Transactions*, vol. 106, pp. 382–391, 2020.

[104] C. A. Ernesto and R. T. Farouki, “High-speed cornering by cnc machines under prescribed bounds on axis accelerations and toolpath contour error,” *The International Journal of Advanced Manufacturing Technology*, vol. 58, no. 1, pp. 327–338, 2012.

[105] S. Tajima, S. Iwamoto, and H. Yoshioka, “Kinematic tool-path smoothing for 6-axis industrial machining robots,” *International Journal of Automation Technology*, vol. 15, no. 5, pp. 621–630, 2021.

[106] M. Riley and C. G. Atkeson, “Robot catching: Towards engaging human-humanoid interaction,” *Autonomous Robots*, vol. 12, no. 1, pp. 119–128, 2002.

[107] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, “Trajectory planning for optimal robot catching in real-time,” in *IEEE International Conference on Robotics and Automation*, pp. 3719–3726, 2011.

[108] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, “Toward a dynamical pick and place,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 2, pp. 292–297, 1995.

[109] N. Uchiyama, S. Sano, and K. Ryuman, “Control of a robotic manipulator for catching a falling raw egg to achieve human-robot soft physical interaction,” in *21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 777–784, 2012.

[110] G. Bätz, A. Yaqub, H. Wu, K. Kühnlenz, D. Wollherr, and M. Buss, “Dynamic manipulation: nonprehensile ball catching,” in *18th Mediterranean Conference on Control and Automation*, pp. 365–370, 2010.

[111] M. M. Schill, F. Gruber, and M. Buss, "Quasi-direct nonprehensile catching with uncertain object states," in *IEEE International Conference on Robotics and Automation*, pp. 2468–2474, 2015.

[112] L. Glomb, F. Liers, and F. Rösel, "A rolling-horizon approach for multi-period optimization," *European Journal of Operational Research*, vol. 300, no. 1, pp. 189-206, 2022.

[113] G. Zhou, C. Zhang, F. Lu, and J. Zhang, "Integrated optimization of cutting parameters and tool path for cavity milling considering carbon emissions," *Journal of Cleaner Production*, vol. 250, 2020.

[114] S. Debnath, M. M. Reddy, and Q. S. Yi, "Influence of cutting fluid conditions and cutting parameters on surface roughness and tool wear in turning process using taguchi method," *Measurement*, vol. 78, pp. 111–119, 2016.

[115] E. Dyllong and A. Visioli, "Planning and real-time modifications of a trajectory using spline techniques," *Robotica*, vol. 21, no. 5, pp. 475–482, 2003.

Publication list

The contents of this thesis are the results of original research and have not been submitted for a higher degree to any other university or institution.

Much work presented in this thesis has been published as journal or conference papers.The following are the list of these papers:

International journal papers

- M. S. Paing, E. W. Nshama, and N. Uchiyama, " A kinematically constrained reparameterization approach to optimal time and jerk motion of industrial machines," *IEEE Access*, vol. 9, pp. 97843-97854, 2021 (Impact factor - 3.476).
- M. S. Paing and N. Uchiyama, " A spline-based approach to smooth and time-optimal trajectory generation of CNC machines with guaranteed kinematic constraints," *The International Journal of Advanced Manufacturing Technology*, vol. 121, pp. 3385-3398, 2022 (Impact factor - 3.5633).

International conference papers

- M. S. Paing, E. W. Nshama, and N. Uchiyama, "Motion trajectory estimation of a flying object and optimal reduced impact catching by a planar manipulator," in *29th IEEE International Conference on Robot and Human Interactive Communication*, pp. 920-925, 2020.

- M. S. Paing and N. Uchiyama, “Spline-based time-optimal control for smooth trajectory generation of CNC machines with geometric constraints,” in *31st IEEE International Symposium on Industrial Electronics*, pp. 1088-1093, 2022.