

Multi-Objective Optimization Approach to Energy-Saving Motion
Trajectory Generation with High-Speed and High-Accuracy for
Industrial Feed Drive Systems

(産業機械送り駆動系の高速高精度化と省エネルギー化のため
の多目的最適化による動作軌道生成)

July, 2021

Doctor of Philosophy (Engineering)

Enock William Nshama

エノックウィリアムンシャマ

Toyohashi University of Technology

Declaration of Authorship

I, Enock William Nshama, declare that this thesis titled, “ Multi-Objective Optimization Approach to Energy-Saving Motion Trajectory Generation with High-Speed and High-Accuracy for Industrial Feed Drive Systems” 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 2021

Abstract

The ever-growing breakthroughs in science and technology perpetually pressurize the manufacturing industry to increase accuracy and production rates in processes. Such increases come at a price, the price of an enlarging carbon footprint with limited reliable energy sources. Computer numerical control (CNC) machine tools are typically used in the industry due to their accuracy and repetitive task execution speed. Due to the afore-mentioned production and energy pressures, the objectives of improving accuracy, cycle time, and energy consumption are major research drivers. Since these objectives are contradictory, Pareto optimization methods are necessary to obtain optimal operating conditions.

There are typically three possible actions for achieving the aforementioned objectives: hardware upgrades, internal software modifications, and trajectory optimizations. Hardware upgrades are usually avoided since they are relatively expensive. Most commercial CNC machine tools' internal software is inaccessible; hence trajectory optimization presents a feasible and cost-effective action. With this rationale, trajectory optimization is the study field explored in this thesis.

This thesis discusses several propositions for trajectory optimization in industrial feed drive systems: Pareto optimization of energy and tolerance in motion trajectory generation (Chapter 3), a trade-off between energy saving and cycle time reduction by Pareto-optimal corner smoothing (Chapter 4), and Pareto optimization of cycle time and motion accuracy (Chapter 5). Feed drive dynamics, energy modeling, trajectory profiling, a multi-objective optimization problem (MOOP) formulation, and a Pareto frontier generation algorithm are described in the preliminaries chapter (Chapter 2). The thesis is completed with a conclusion and future works chapter (Chapter 6).

A method of generating piecewise linear trajectories with smoothed corners optimizing two objectives: energy consumption and cornering tolerance for feed drive systems is proposed in Chapter 3. An energy model of an industrial biaxial feed drive system is used to formulate a bi-objective optimization problem (BOOP). The linear and smooth corner segments are respectively described using jerk-limited acceleration profiles (JLAPs) and kinematic corner smoothing with interrupted accelerations (KCSIA). The optimization problem is formulated with the normalized normal constraints method, where sequential quadratic programming is used to solve it. A divide and conquer algorithm is utilized to generate Pareto optimal solutions recursively. The best trade-off solution is obtained as the one that minimizes both objectives. Optimization results for an industrial biaxial machine are illustrated, where the best trade-off solution achieves $\sim 64\%$ of the energy-saving potential with a moderate cornering tolerance of $\sim 30\ \mu\text{m}$.

Chapter 4 proposes a method of generating Pareto optimal corner smoothing trajectories that trade-off the contradicting objectives of minimizing cycle time and energy consumption. Several studies have proposed corner smoothing methods that improve cycle time for piecewise linear paths by exploiting axial limits to achieve time-optimal trajectories. Energy-saving is not considered an objective in these methods. The trajectories along linear paths and smoothed corners are respectively described using JLAP and kinematic corner smoothing (KCS) methods (i.e., KCSIA and kinematic corner smoothing with uninterrupted acceleration (KCSUA)). An energy consumption model of an industrial two-axis feed drive system is identified by least squares estimation (LSE) and used in solving the bi-objective optimization problem (BOOP). A contrast and comparison are made between KCSIA, KCSUA, and point-to-point (PTP) motion profiles. The optimization results show that the KCSIA Pareto frontier is closest to the utopia point, where it is experimentally vindicated that the best trade-off trajectory achieves $\sim 66\%$ and $\sim 60\%$ of the time and energy-saving potentials, respectively. In terms of contouring performance of best trade-off trajectories, while KCSUA reduces the average error by $\sim 7\%$, KCSIA decreases the maximum error by $\sim 19\%$ relative to PTP.

A method of Pareto optimizing the conflicting objectives of reducing cycle time and increasing cornering accuracy for piecewise linear contours is proposed in Chapter 5. It has been shown in the literature that non-zero cornering velocities deteriorate contouring performance while reducing cycle time. To resolve the set of conflicting objectives, the normalized normal constraint (NNC) formulation of the BOOP is described with cornering tolerances at each corner

described as inequality constraints. This method's effectiveness is investigated with linear and smoothed corner segments, respectively, defined by JLAP and KCSIA. The optimized KCSIA is referred to as KCSIA*, where its Pareto frontier shows that the original KCSIA produces a dominated solution. Hence, KCSIA* solutions are superior compared to KCSIA. Experimental results further emphasize this point by showing that the KCSIA* had a lower contouring error than KCSIA, where the best trade-off solution reduces the maximum and average contouring errors by $\sim 29\%$ and $\sim 12\%$ while increasing cycle time by $\sim 3\%$ compared to KCSIA.

Concluding remarks and tentative future works of this thesis are illustrated in Chapter 6. The proposed methods can be extended to five-axis CNC machines by including tool orientation tolerances at corners. In the case of machining operations, cutting forces increase the amount of energy consumption. Hence, a cutting force model can be incorporated into the previously used energy consumption model. The proposed methods can be generalized by considering asymmetrical corner smoothing together with cornering transitions other than line-to-line ones.

Acknowledgments

I would like to express my great appreciation to my supervisor, Prof. Naoki Uchiyama for his patient and supportive guidance during my study. Under his tutelage, I was not only able to enhance my knowledge in the field of research but also I learned the methodology, teamwork and ethics related to scientific study. This acquired knowledge and skills shall be beneficial to me in my career.

I take this opportunity to thank my committee members: Prof. Kentaro Takagi and Prof. Yohei Abe for their time, effort and expertise in controlling the quality of my research study. I would like also to express my gratitude to all members of the System Engineering Laboratory for their technical support and companionship during my study.

Furthermore, I would like to thank Monbukagakusho for providing me a scholarship during my study. I wish to extend my appreciation to all staff members of the Toyohashi University of Technology for their interminable assistance and friendship. My gratitude goes to my parent university, the University of Dar es Salaam, for their support and patience during my study.

Last but not the least, I would like to express my deep gratitude to my family for their unconditional support and encouragement throughout my study.

Contents

Abbreviations

List of Figures

List of Tables

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Thesis contributions and outline | 3 |
| 1.2.1 | Contributions | 3 |
| 1.2.2 | Outline | 4 |
| 2 | Preliminaries and Literature Review | 7 |
| 2.1 | Biaxial feed drive systems | 7 |
| 2.1.1 | System dynamics | 7 |
| 2.1.2 | Motion accuracy | 9 |
| 2.1.3 | Experimental system setup | 10 |
| 2.2 | Energy consumption model | 13 |
| 2.3 | Jerk-limited acceleration profiles | 15 |
| 2.4 | Corner smoothing methods | 17 |
| 2.4.1 | Kinematic corner smoothing | 18 |
| 2.4.1.1 | Kinematic corner smoothing with interrupted acceleration | 19 |
| 2.4.1.2 | Kinematic corner smoothing with uninterrupted acceleration | 21 |
| 2.5 | Multi-objective optimization methods | 23 |
| 2.6 | Divide and conquer algorithm | 29 |
| 2.7 | Global Pareto-optimal filter | 30 |
| 3 | Pareto Optimization of Energy and Tolerance in Motion Trajectory Generation | 33 |
| 3.1 | Introduction | 33 |
| 3.2 | Trajectory representation | 34 |

| | | |
|----------|---|-----------|
| 3.2.1 | Jerk-limited acceleration profiles | 35 |
| 3.2.2 | Kinematic corner smoothing with interrupted acceleration | 36 |
| 3.3 | Trajectory optimization | 38 |
| 3.3.1 | Problem formulation | 38 |
| 3.3.2 | Trajectory constraints | 40 |
| 3.4 | Optimization results and discussions | 42 |
| 3.5 | Summary | 45 |
| 4 | A Trade-off between Energy Saving and Cycle Time Reduction by Pareto-optimal Corner Smoothing | 47 |
| 4.1 | Introduction | 47 |
| 4.2 | Identification of the energy consumption model | 51 |
| 4.2.1 | Model identification | 51 |
| 4.2.2 | Identification results | 52 |
| 4.2.3 | Model verification | 55 |
| 4.3 | Trajectory representation | 56 |
| 4.3.1 | Jerk-limited acceleration profile | 56 |
| 4.3.2 | Kinematic corner smoothing | 58 |
| 4.3.2.1 | Kinematic corner smoothing with interrupted acceleration | 59 |
| 4.3.2.2 | Kinematic corner smoothing with uninterrupted acceleration | 61 |
| 4.4 | Trajectory optimization | 62 |
| 4.4.1 | Problem formulation | 63 |
| 4.4.2 | Optimization constraints | 65 |
| 4.4.3 | Pareto generation | 66 |
| 4.4.4 | Space complexity analysis | 67 |
| 4.5 | Results and discussions | 69 |
| 4.5.1 | Optimization results | 70 |
| 4.5.2 | Simulation results | 71 |
| 4.5.3 | Experimental results | 73 |
| 4.6 | Summary | 77 |
| 5 | Pareto Optimization of Cycle Time and Motion Accuracy in Trajectory Planning for Industrial Feed Drive Systems | 79 |
| 5.1 | Introduction | 79 |
| 5.2 | Pareto-optimal trajectory design | 83 |
| 5.2.1 | Problem formulation | 83 |
| 5.2.2 | Optimization constraints | 87 |
| 5.2.3 | Pareto generation | 88 |
| 5.2.4 | Trajectory representation | 89 |
| 5.2.4.1 | Jerk-limited acceleration profile | 89 |

| | | |
|----------|--|------------|
| 5.2.4.2 | Kinematic corner smoothing with interrupted acceleration . . . | 91 |
| 5.2.4.3 | Space complexity analysis | 95 |
| 5.3 | Results and discussions | 95 |
| 5.3.1 | Optimization conditions | 95 |
| 5.3.2 | Case I: Star-shaped tool path | 96 |
| 5.3.2.1 | Optimization results | 97 |
| 5.3.2.2 | Experimental results | 99 |
| 5.3.3 | Case II: Complex tool path | 101 |
| 5.3.3.1 | Optimization results | 101 |
| 5.3.3.2 | Experimental results | 103 |
| 5.4 | Summary | 106 |
| 6 | Conclusions and Future Works | 109 |
| 6.1 | Conclusions | 109 |
| 6.2 | Future works | 110 |
| | Bibliography | 113 |
| | Publication List | 125 |

Abbreviations

| | |
|--------------|--|
| AC | alternating current |
| BOOP | bi-objective optimization problem |
| CAD | computer-aided design |
| CAM | computer-aided manufacturing |
| CHIM | convex hull of individual minima |
| CNC | computer numerical control |
| CPU | central processing unit |
| EMF | electromotive force |
| FDS | feed drive system |
| FIR | finite impulse response |
| JLAP | jerk-limited acceleration profile |
| KCS | kinematic corner smoothing |
| KCSIA | kinematic corner smoothing with interrupted acceleration |
| KCSUA | kinematic corner smoothing with uninterrupted acceleration |
| LSE | least squares estimation |

| | |
|--------------|---------------------------------------|
| MOOP | multi-objective optimization problem |
| NURBS | non-uniform rational B-spline |
| NBI | normal boundary intersection |
| NNC | normalized normal constraint |
| PD | proportional-derivative |
| PTP | point-to-point |
| SQP | sequential quadratic programming |
| SOOP | single-objective optimization problem |
| RAM | random access memory |
| RMS | root mean square |

List of Figures

| | | |
|------|--|----|
| 1.1 | CNC machine tools | 2 |
| 1.2 | A block diagram of a typical CNC machine tool. | 2 |
| 1.3 | Thesis outline. | 6 |
| 2.1 | Biaxial feed drive system applications | 8 |
| 2.2 | A demonstration of tracking and contouring errors at time instant t | 10 |
| 2.3 | The schematic illustration of the industrial biaxial table used for experimental verification. | 11 |
| 2.4 | Industrial biaxial feed drive systems graphical user interface. | 12 |
| 2.5 | An exemplary description of JLAPs. | 15 |
| 2.6 | A demonstration of the difference between global and local corner smoothing methods. | 17 |
| 2.7 | Geometric description of a KCS curve. | 19 |
| 2.8 | Exemplary jerk, acceleration and velocity profiles generated by KCSIA. | 20 |
| 2.9 | Exemplary jerk, acceleration and velocity profiles generated by KCSUA. | 22 |
| 2.10 | An illustration of mapping a candidate solution in a feasible region of decision space to/from an objective space. The shaded area marks the feasible region. The Pareto frontier is mark as a solid red curve. \mathbf{O} marks the utopia point. | 24 |
| 2.11 | An illustration of global and local Pareto frontiers of a concave feasible region Ω | 25 |
| 2.12 | A demonstration of the ϵ -constraint method. | 26 |
| 2.13 | A demonstration of the normal boundary intersection (NBI) method. | 27 |
| 2.14 | A demonstration of the NNC method. | 28 |
| 2.15 | Illustration of the divide and conquer algorithm for a two-objective case. | 30 |
| 3.1 | Geometric description of the KCSIA approach. | 36 |
| 3.2 | Contour used for optimal trajectory generation. | 43 |
| 3.3 | Pareto front representation of optimization results | 44 |
| 3.4 | Generated velocity profiles at different optimal weighting factors. | 44 |
| 3.5 | Variation of best trade-off cornering tolerance and energy saving potential under upper bound of cornering tolerance. | 45 |
| 4.1 | Reference acceleration and velocity profiles used for the identification experiment. | 53 |

| | | |
|------|--|----|
| 4.2 | Measured and estimated power consumption during the identification experiment. The measured power is labeled as ‘Measured’ while the power estimated by the model is labeled as ‘Estimated’. | 54 |
| 4.3 | Measured and estimated energy consumption during identification experiment. The energy computed from power measurements and model estimates are labeled as ‘Measured’ and ‘Estimated’, respectively. | 54 |
| 4.4 | Measured and estimated power consumption during the verification experiment. The measured power is labeled as ‘Measured’ while the power computed by the model is labeled as ‘Estimated’. | 55 |
| 4.5 | Measured and estimated energy consumption during verification experiment. The energy computed from power measurements and model estimates are labeled as ‘Measured’ and ‘Estimated’, respectively. | 56 |
| 4.6 | Geometric description of a cornering path using the KCS method. | 58 |
| 4.7 | An illustration of geometric constraints (4.43)–(4.45) imposed on the optimization problem (4.37). | 67 |
| 4.8 | The piecewise linear contours (black dotted lines) used for optimal trajectory generation. An illustration of corner smoothed reference trajectories generated using KCSUA (green dash-dotted lines) and KCSIA (blue dashed lines) are shown. | 70 |
| 4.9 | Optimization results represented as a Pareto frontier. | 71 |
| 4.10 | The generated jerk, acceleration and velocity profiles for the x_1 axis. | 72 |
| 4.11 | The generated jerk, acceleration and velocity profiles for the x_2 axis. | 73 |
| 4.12 | The generated tangential velocities for time, best trade-off and energy optimality scenarios. | 74 |
| 4.13 | Curvature of the generated trajectories for time, best trade-off and energy optimality scenarios. | 74 |
| 4.14 | Simulation results of contouring performance. | 75 |
| 4.15 | Experimental results represented as a Pareto frontier. The time-optimal, best trade-off and energy-optimal experimental results correspond to the grey-dashed ellipses respectively labeled as 1*, 2* and 3*. | 76 |
| 4.16 | Experimental results of contouring performance. | 77 |
| 4.17 | Experimental contouring performance at different weighting factors. | 78 |
| 5.1 | An illustration of a cornering path generated by a corner smoothing method. A corner point \mathbf{p}_c is smoothed by a red dash-dot curve with a cornering error ε bounded by $0 < \varepsilon \leq \varepsilon_{ub}$. | 84 |
| 5.2 | An illustration of the reformulated optimization problem (5.9) with the feasible region reduced by the normalized normal constraint inequality constraint (5.10). The utopia point is marked by the origin \mathbf{O} . | 87 |
| 5.3 | The star-shaped tool path used for Pareto-optimal trajectory generation. The motion direction is shown by the arrow. Details 1 and 2 show the cornering motions at different weighting factors ζ for the acute and obtuse corners, respectively. | 96 |

| | | |
|------|--|-----|
| 5.4 | Optimization results of the star-shaped tool path represented as a Pareto frontier. | 97 |
| 5.5 | An illustration of Pareto optimal jerk, acceleration and velocity profiles of the star-shaped tool path corresponding to different weighting factors ζ , where kinematic constraints (5.14) are obeyed $\forall \zeta$. | 98 |
| 5.6 | Tangential velocities and cornering errors of Pareto optimal trajectories for the star-shaped tool path at different weighting factors ζ , where ε is within the cornering constraints (5.15). | 99 |
| 5.7 | The 7 th trial contouring error performance for the star-shaped tool path. The error tends to be within the set cornering tolerance $0 < \varepsilon \leq 200 \mu\text{m}$ | 101 |
| 5.8 | Contouring performance experimental results for the star-shaped tool path. | 101 |
| 5.9 | Experimental contouring performance for the star-shaped tool path. | 102 |
| 5.10 | The butterfly-shaped path tool path used to represent complex paths in trajectory generation. The motion direction is shown by the dashed arrow. | 103 |
| 5.11 | Optimization results for the complex tool path represented as a Pareto frontier. | 104 |
| 5.12 | An illustration of Pareto-optimal jerk, acceleration and velocity profiles of the complex tool path corresponding to different weighting factors ζ , where kinematic constraints (5.14) are obeyed $\forall \zeta$. | 104 |
| 5.13 | Cornering errors of Pareto-optimal trajectories of the complex tool path at different weighting factors ζ , where ε is within the cornering constraints (5.15). | 105 |
| 5.14 | The 3 rd trial contouring error performance for the complex tool path. The error tends to be within the set cornering tolerance $0 < \varepsilon \leq 200 \mu\text{m}$ | 105 |
| 5.15 | Contouring performance experimental results for the complex tool path. | 106 |
| 5.16 | Experimental contouring performance for the complex tool path. | 107 |

List of Tables

| | | |
|-----|---|-----|
| 2.1 | Identified plant parameters | 12 |
| 3.1 | Identified energy consumption model parameters | 42 |
| 4.1 | Identified energy coefficients | 53 |
| 4.2 | Summary of optimization results for the best trade-off trajectories | 71 |
| 4.3 | Experimental results summary of energy consumption for the best trade-off trajectories | 77 |
| 4.4 | Summary of experimental results of contouring performance for the best trade-off trajectories | 77 |
| 5.1 | Results summary of Pareto-optimal solutions for the star-shaped tool path . . . | 97 |
| 5.2 | Influence the minimum trade-off level on the optimization process | 99 |
| 5.3 | Experimental results summary of contouring performance for the star-shaped tool path | 100 |
| 5.4 | Results summary of the Pareto optimal solutions for the complex tool path . . . | 102 |
| 5.5 | Experimental results summary of contouring performance for the complex tool path | 106 |

Chapter 1

Introduction

1.1 Motivation

The manufacturing industry is a key player in the global economy since it supplies products to an ever-growing consumer base. Product miniaturization demands over the last 50 years have led to the evolution of manufacturing technology from man-operated to computer numerical control (CNC) machine tools (Fig. 1.1) [15]. Currently, CNC machine tools are widely used due to their speed, flexibility, accuracy and precision in production processes.

Fig. 1.2 shows a general block diagram of a CNC machine tool. Product geometrical information is converted to a reference trajectory (i.e., motion commands) using computer-aided design (CAD)/computer-aided manufacturing (CAM) software and then stored in a machine control unit. Via an in-built controller, the reference trajectory is tracked by a feed drive system. A feed drive system typically consists of a rotary/linear servomotor that produces torque/force for actuating motions along an axis.

Currently, there is a global concern in reducing energy consumption in the industrial sector [68]. The manufacturing industry, spearheaded by machine tools, is a major consumer of electrical energy [5, 78, 107]. This has provoked a challenge of improving the quality and quantity of products in adherence to environmental sustainability. This challenge is manifested as a set



(a) Milling machine. <http://www.datancnc.com/> (b) Lathe machine. <https://www.americamachinetools.com/>

Fig. 1.1: CNC machine tools

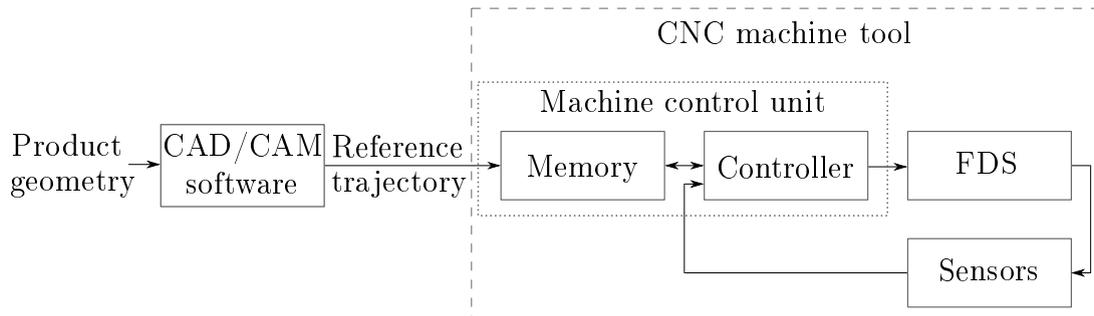


Fig. 1.2: A block diagram of a typical CNC machine tool.

of feed drive system performance objectives: saving energy, cycle time reduction and motion accuracy improvement.

Controller design strategies are proposed in the literature for energy-saving [26, 61] and improving accuracy [13, 14, 94]. Such strategies are limited by the accessibility of control units in in-service CNC machinery. Reference trajectory generation methods have been extensively studied due to their applicability in in-service feed drive system. Energy-saving [35, 40, 65, 66], cycle time minimizing [10, 29, 44, 57, 71] and accuracy improving [38, 49, 75] trajectory generators have been studied. The existing methods are unable to simultaneously address the

objectives due to the contradictory nature of the aforementioned objectives [24, 30, 72, 85]. Thus, this thesis aims to develop trajectory generation approaches that provide Pareto-optimal solutions for the feed drive system performance objectives.

1.2 Thesis contributions and outline

1.2.1 Contributions

The contradictory objectives of energy-saving, cycle time reduction and motion accuracy improvement are the main research drivers in the manufacturing industry. In the literature, various researchers have proposed trajectory generation methods that only tackle one of these objectives. In order to address these objective contradictions, Pareto optimization methods of trajectory generation for industrial feed drive systems are contributed as follows:

- The focus of corner smoothing methods in the literature has been reducing cycle time by maximizing cornering velocities under user-specified tolerances. There is a gap in the literature regarding corner smoothing under fixed cycle time conditions. A corner smoothing method that provides a trade-off between energy consumption and cornering error at predefined cycle times is proposed in Chapter 3. This method has practical advantages in two-fold over existing corner smoothing approaches: it can generate corner smoothed trajectories at different cycle times (i.e., optimal or otherwise) which is useful in supply chain economic scheduling and it offers Pareto-optimal solutions of energy consumption and cornering error at a selected cycle time.
- Several studies have proposed corner smoothing methods that improve cycle time for piecewise linear paths without energy consumption considerations. Chapter 4 presents a method of generating Pareto-optimal local corner smoothing trajectories that compromise the contradicting objectives of minimizing cycle time and energy consumption. In order to model the energy consumption of an industrial feed drive system, a least square error-based model identification is proposed and verified. Optimization and experimental results show that corner smoothing not only improves cycle time but also reduces energy

consumption. The proposed corner smoothing method has a practical advantage over existing approaches in the fact that it can provide Pareto-optimal solutions to energy-saving and cycle time reduction objectives.

- Existing corner smoothing approaches reduce cycle time, at the expense of motion accuracy, by maximizing cornering velocity. A Pareto-optimal local corner smoothing method that compromises cycle time with motion accuracy at corners is proposed in Chapter 5. A kinematic corner smoothing approach, from the literature, is modified in order to achieve Pareto-optimality. Results show that the time-optimal solution of the proposed method has a shorter cycle time and higher motion accuracy than that of the kinematic corner smoothing approach. The proposed method has a practical advantage over other existing approaches since it offers trade-off solutions to cycle time and motion accuracy objectives.

1.2.2 Outline

The section describes brief descriptions of the following chapters in this thesis:

- Chapter 2 introduces models, fundamental algorithms and methods used throughout this research. Dynamics and energy consumption models of biaxial feed drive systems are presented followed by a description of an industrial feed drive system experimental setup. Jerk-limited acceleration profiles (JLAPs) and corner smoothing methods reviews are subsequently represented. Afterward, a review of multi-objective optimization methods is described. Finally, the divide and conquer algorithm and global Pareto-optimal filter algorithms are presented.
- Chapter 3 proposes a method of generating fixed cycle time piece-wise linear trajectories with locally smoothed corners that optimizes two objectives: energy consumption and cornering tolerance for feed drive systems. An energy model of an industrial biaxial feed drive system is used to formulate a bi-objective optimization problem (BOOP). The linear and smooth corner segments are generated using JLAPs and kinematic corner smoothing with interrupted acceleration (KCSIA), respectively. The optimization problem is described with normalized normal constraint (NNC), where sequential quadratic programming (SQP) is used to solve it. The divide and conquer algorithm is utilized

to recursively generate Pareto optimal solutions. Optimization results for an industrial bi-axial feed drive system are illustrated, where the contradiction between energy-saving and corner smoothing is validated and the best trade-off solution selected as the one that minimizes both objectives.

- Chapter 4 presents Pareto-optimal local corner smoothing trajectory generator to trade-off the contradicting objectives of minimizing cycle time and energy consumption. Piecewise linear contours are considered, where motions along linear segments and smoothed corners are respectively described using JLAPs and kinematic corner smoothing (KCS) approaches. A least squares estimation (LSE)-based energy consumption model identification approach is demonstrated and used to model an industrial bi-axial feed drive system. The resulting model used in solving the BOOP. From the resulting Pareto frontier, the best trade-off trajectory is selected as the one that minimizes both objectives. The presented method's effectiveness is vindicated via simulations and experiments, where the best trade-off trajectory maximizes the time and energy-saving potentials.
- Chapter 5 proposes a Pareto-optimal local corner smoothing method to trade-off between cycle time and motion accuracy in trajectory planning for industrial feed drive systems (FDSs). The objective of motion accuracy improvement is represented by a corner smoothing minimization criterion in the BOOP. Linear and cornering motions along a contour are respectively described by JLAPs and a modified KCSIA approach. A Pareto frontier is generated by the divide and conquer algorithm with the best trade-off solution selected as the one closest to the utopia point. The effectiveness of the proposed method is validated through simulations and experiments.
- Chapter 6 summarizes the key research findings discussed in this thesis. It also identifies uncharted study avenues for improving the presented methods and guiding potential future works.

The chapters are organized as illustrated in Fig. 1.3.

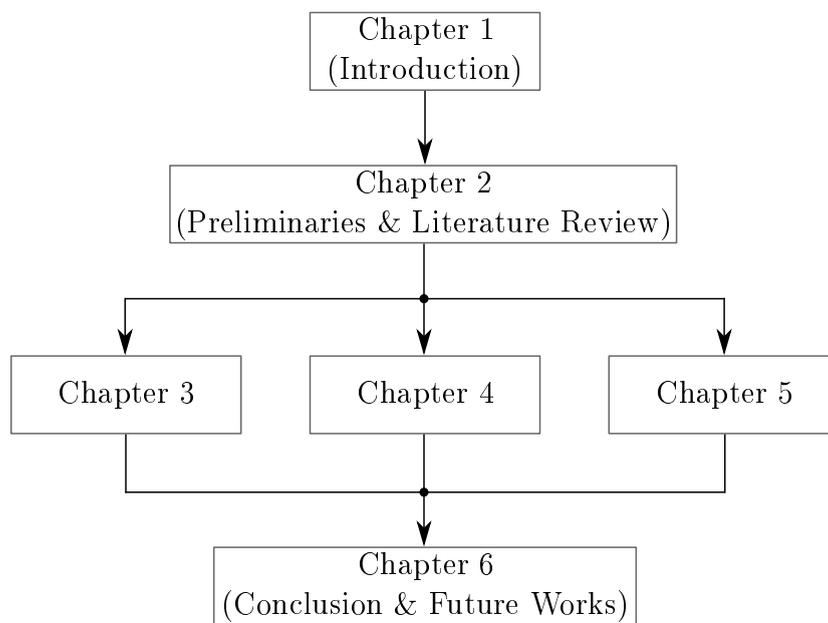


Fig. 1.3: Thesis outline.

Chapter 2

Preliminaries and Literature Review

2.1 Biaxial feed drive systems

2.1.1 System dynamics

Biaxial feed drive systems are commonly utilized for tracking planar motions in the manufacturing industry, where they appear in machineries such as water-jet cutting, wire electrical discharge and laser cutting machines (Fig. 2.1). The dynamics of a conventional biaxial feed drive system can be represented by the a second order decoupled model [12–14, 61, 67]

$$\mathbf{M}_t \ddot{\mathbf{x}}(t) + \mathbf{D}_t \dot{\mathbf{x}}(t) + \mathbf{F}_c \text{sgn}(\dot{\mathbf{x}}(t)) = \mathbf{f}(t), \quad (2.1)$$

with

$$\mathbf{M}_t = \text{diag} \{m_{t,k}\}, \quad \mathbf{D}_t = \text{diag} \{d_{t,k}\}, \quad \mathbf{F}_c = \text{diag} \{f_{c,k}\}, \quad k = \{1, 2\},$$

where $m_{t,k}$ is the k^{th} axis inertia. $f_{c,k}$ and $d_{t,k}$ are the Coulomb and translational viscous frictions, respectively. $\text{sgn} \{\dot{\mathbf{x}}\} \in \mathbb{R}^2$ is a vector whose components are the signs of the respective components of $\dot{\mathbf{x}}$. $\mathbf{f} \in \mathbb{R}^2$ and $\mathbf{x} \in \mathbb{R}^2$ are the driving force and axial position vectors, respectively. Assuming each axis is actuated by a servomotor via a ball screw, the following motor



(a) Wire electrical discharge machine. <http://www.jsedm.com/>
 (b) Water-jet cutting machine. <https://www.headwaterjet.net/>



(c) Laser cutting machine. <https://www.xtlaser.com/>

Fig. 2.1: Biaxial feed drive system applications

dynamics

$$\mathbf{H}\ddot{\boldsymbol{\theta}}(t) + \mathbf{D}_r\dot{\boldsymbol{\theta}}(t) + \boldsymbol{\tau}(t) = \mathbf{K}_r\mathbf{i}(t), \quad (2.2)$$

with

$$\mathbf{H} = \text{diag}\{h_k\}, \quad \mathbf{D}_r = \text{diag}\{d_{r,k}\}, \quad \mathbf{K}_r = \text{diag}\{k_{r,k}\}$$

are considered, where h_k , $d_{r,k}$ and $k_{r,k}$ are the respective motor inertia, rotary viscous frictions and torque constant for the k^{th} axis. $\boldsymbol{\tau} \in \mathbb{R}^2$ and $\mathbf{i} \in \mathbb{R}^2$ are the torque and current vectors, respectively. For ball screws, rotary and translational motions are correlated by

$$\mathbf{f}(t) = 2\pi\mathbf{\Gamma}^{-1}\boldsymbol{\tau}(t), \quad \mathbf{x}(t) = \frac{1}{2\pi}\mathbf{\Gamma}\boldsymbol{\theta}(t), \quad (2.3)$$

with $\mathbf{\Gamma} = \text{diag}\{\gamma_k\}$, where γ_k is the ball screw lead. Hence, the overall dynamics of a biaxial feed drive system are obtained by combining (2.1)–(2.3) to obtain

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{D}\dot{\mathbf{x}}(t) + \mathbf{F}_c \text{sgn}(\dot{\mathbf{x}}(t)) = \mathbf{K}_F \mathbf{i}(t), \quad (2.4)$$

with

$$\begin{aligned} \mathbf{M} &= \text{diag}\{m_k\}, & \mathbf{D} &= \text{diag}\{d_k\}, & \mathbf{K}_F &= \text{diag}\{k_{F,k}\} \\ m_k &= m_{\iota,k} + \frac{4\pi^2}{\gamma_k^2} h_k, & d_k &= d_{\iota,k} + \frac{4\pi^2}{\gamma_k^2} d_{\tau,k} & k_{F,k} &= \frac{2\pi}{\gamma_k} k_{\tau,k}. \end{aligned}$$

2.1.2 Motion accuracy

Accuracy is a vital criterion in assessing a machine tools' performance. It is evaluated based on error measurements that are acquired as the machine tool follows a reference contour. A machine tool's motion accuracy is subdivided into positional and contouring accuracies which are respectively evaluated by tracking and contouring error measurements [16, 43].

Tracking error refers to the difference between reference and actual positions along each feed drive system axis. Contouring error is defined as the minimum distance between a reference contour and an actual position [12, 60, 99]. Tracking and contouring errors are illustrated in Fig. 2.2.

In a manufacturing process, product features are converted into reference contours by a CAD/CAM framework. Since contouring errors show the deviation of a machine toolpath from a reference contour, it is vital to monitor them in order to ensure the machine tool's motion is within specified accuracy tolerances [13, 14, 30]. In machining, product accuracy is generally affected by workpiece fixation errors, tool wear and contouring errors [46]. Since reference trajectory generation algorithms can only improve product accuracy by mitigating contouring errors, proposed algorithms' performances are compared via contouring error measurements in the literature [30, 42, 43, 84, 85, 97]. Thus, accuracy improvement performances of the presented algorithms, in this thesis, are assessed using contouring error measurements.

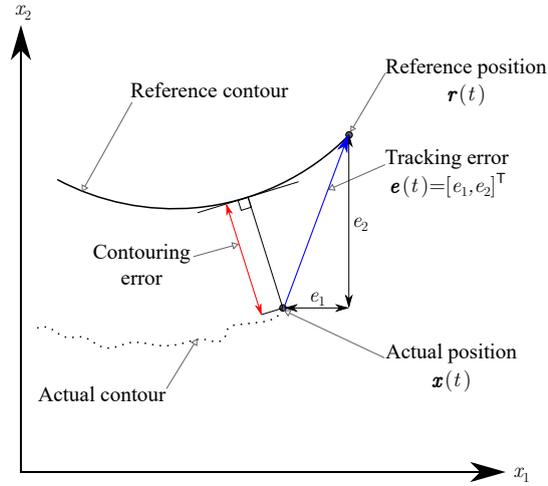


Fig. 2.2: A demonstration of tracking and contouring errors at time instant t .

2.1.3 Experimental system setup

In this thesis, biaxial feed drive systems are represented by an industrial biaxial table shown in Fig. 2.3. Motion along each axis is actuated by computer controlled alternating current (AC) rotary servomotors, where linear motion is acquired via ball screws. 76.29 nm resolution rotary encoders sample table position at a 5 kHz rate. Table velocity is computed as a sampled position numerical difference, where 20 Hz and 60 Hz low-pass filters suppress noise effects for the x_1 and x_2 axes, respectively. Via a graphical user interface (Fig. 2.4), reference trajectories are fed into a desktop computer, having Intel (R) Core i7-3770K central processing unit (CPU), 3.50 GHz, 8 GB random access memory (RAM) and Ubuntu 15.04 64 bit operating system in a Xenomai 3.0 real-time framework, which receives encoder data and sends control signals to the AC rotary servomotors. The experimental system follows a reference trajectory signal stream at a 5 kHz rate and terminates an experiment when the stream ends. The total motion duration of the reference trajectory is stored as the experiment's cycle time. Electric power is measured, at a 50 ms data update interval, by a HIOKI 3390 power analyzer using the two wattmeter method [1].

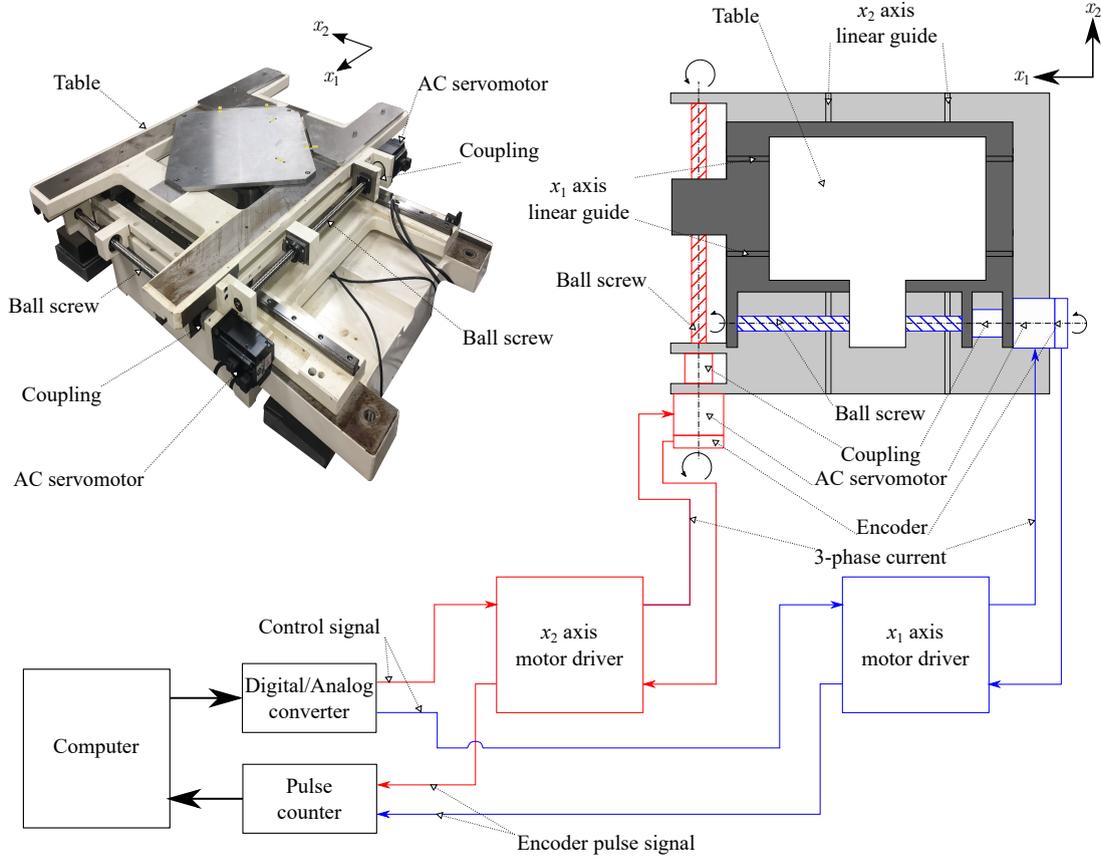


Fig. 2.3: The schematic illustration of the industrial biaxial table used for experimental verification.

In a similar manner to [59], the tracking error dynamics

$$\begin{aligned} \mathbf{e}(t) &= \mathbf{r}(t) - \mathbf{x}(t), \\ \ddot{\mathbf{e}}(t) &= \ddot{\mathbf{r}}(t) + \mathbf{M}^{-1} \left\{ \mathbf{u}(t) - \mathbf{D}\dot{\mathbf{x}}(t) - \mathbf{F}_c \text{sgn} \{ \dot{\mathbf{x}}(t) \} \right\} \end{aligned} \quad (2.5)$$

are described, where $\mathbf{r}(t) \in \mathbb{R}^2$ and $\mathbf{u}(t) \in \mathbb{R}^2$ are the reference trajectory and control input vectors, respectively. A proportional-derivative (PD) tracking controller with Coulomb and viscous friction compensation

$$\begin{aligned} \mathbf{u}(t) &= \hat{\mathbf{M}} \left\{ \ddot{\mathbf{r}}(t) + \mathbf{K}_P \mathbf{e}(t) + \mathbf{K}_D \dot{\mathbf{e}}(t) \right\} + \hat{\mathbf{D}} \dot{\mathbf{x}}(t) + \hat{\mathbf{F}}_c \text{sgn} \{ \dot{\mathbf{x}}(t) \} \\ \mathbf{i}(t) &= \mathbf{K}_F^{-1} \mathbf{u}(t) \end{aligned} \quad (2.6)$$

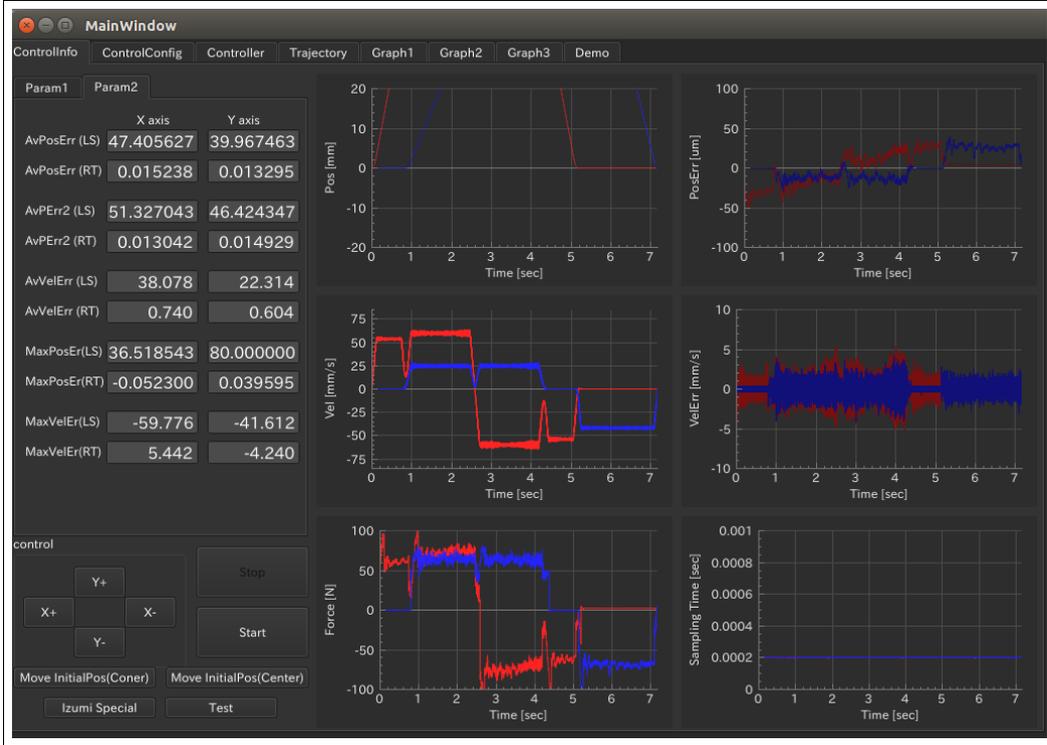


Fig. 2.4: Industrial biaxial feed drive systems graphical user interface.

Table 2.1: Identified plant parameters

| k^{th} axis | $\hat{m}_k \text{Ns}^2/\text{m}$ | $\hat{d}_k \text{Ns}/\text{m}$ | $\hat{f}_k \text{N}$ |
|----------------------|----------------------------------|--------------------------------|----------------------|
| 1 | 86.72 | 558.62 | 47.50 |
| 2 | 99.65 | 795.50 | 58.00 |

is implemented. $\hat{\mathbf{M}} = \text{diag} \{ \hat{m}_k \}$, $\hat{\mathbf{D}} = \text{diag} \{ \hat{d}_k \}$ and $\hat{\mathbf{F}} = \text{diag} \{ \hat{f}_k \}$ correspond to the identified nominal values of \mathbf{M} , \mathbf{D} and \mathbf{F}_c , respectively. $\mathbf{K}_P = \text{diag} \{ k_{P,k} \}$ and $\mathbf{K}_D = \text{diag} \{ k_{D,k} \}$ are the proportional and derivative gain diagonal matrices. The controller in (2.6) is used as the control law for conducting simulations and experiments with identified plant parameters (see Table 2.1), $k_{P,k} = 7225 \text{ s}^{-2}$ and $k_{D,k} = 170 \text{ s}^{-1}$ for $\forall k$.

2.2 Energy consumption model

Similar to the method in [91, 92], energy consumed by a biaxial feed drive system can be modeled. Without any loss in generality, it is assumed that the feed drive system is actuated by AC servomotors. Hence, the electric power consumed by the biaxial feed drive system

$$P(t) = \sqrt{3}\mathbf{u}(t)^\top \mathbf{\Lambda}\mathbf{i}(t) \quad (2.7)$$

is defined with $\mathbf{\Lambda} = \text{diag}\{\lambda_k\}$, where λ_k is the motor power factor for each axis. \mathbf{u} is the root mean square (RMS) motor voltage vector calculated by

$$\mathbf{u}(t) = \mathbf{Z}\mathbf{i}(t) + \mathbf{K}_E\dot{\mathbf{x}}(t), \quad (2.8)$$

with

$$\mathbf{Z} = \text{diag}\{z_k\}, \quad \mathbf{K}_E = \text{diag}\{k_{E,k}\},$$

where \mathbf{Z} and \mathbf{K}_E are the motor impedance and back electromotive force (EMF) constant diagonal matrices, respectively. The combination of (2.4), (2.7) and (2.8) leads to an energy consumption model

$$P(t) = \ddot{\mathbf{x}}(t)^\top \mathbf{C}_1\ddot{\mathbf{x}}(t) + \dot{\mathbf{x}}(t)^\top \mathbf{C}_2\dot{\mathbf{x}}(t) + \dot{\mathbf{x}}(t)^\top \mathbf{C}_3\text{sgn}(\dot{\mathbf{x}}(t)) + \text{tr}(\mathbf{C}_4) + \ddot{\mathbf{x}}(t)^\top \mathbf{C}_5\text{sgn}(\dot{\mathbf{x}}(t)) + \ddot{\mathbf{x}}(t)^\top \mathbf{C}_6\dot{\mathbf{x}}(t), \quad (2.9)$$

$$E = \int_{t_0}^{t_f} P(t) dt,$$

with

$$\begin{aligned} \mathbf{C}_j &= \text{diag}\{c_{j,k}\}, & j &= \{1, 2, \dots, 6\}, \\ c_{1,k} &= \sqrt{3}\lambda_k m_k^2 \frac{z_k}{k_{F,k}^2}, & c_{2,k} &= \sqrt{3}\lambda_k d_k \left(\frac{z_k d_k}{k_{F,k}^2} + \frac{k_{E,k}}{k_{F,k}} \right), \\ c_{3,k} &= \sqrt{3}\lambda_k f_k \left(\frac{2z_k d_k}{k_{F,k}^2} + \frac{k_{E,k}}{k_{F,k}} \right), & c_{4,k} &= \sqrt{3}\lambda_k f_k^2 \frac{z_k}{k_{F,k}^2}, \\ c_{5,k} &= 2\sqrt{3}\lambda_k f_k m_k \frac{z_k}{k_{F,k}^2}, & c_{6,k} &= \sqrt{3}\lambda_k m_k \left(\frac{2z_k d_k}{k_{F,k}^2} + \frac{k_{E,k}}{k_{F,k}} \right), \end{aligned} \quad (2.10)$$

where E is the energy consumed by the biaxial feed drive system during motion from time instant t_0 to t_f . $c_{j,k}$ is the j^{th} energy model coefficient for the k^{th} axis. $\text{tr}(\mathbf{C}_4)$ denotes the trace of matrix \mathbf{C}_4 .

(2.9) can describe the energy consumption of the industrial biaxial table (Fig. 2.3) under a no-load condition. The Coulomb friction acting along the feed drive system axes varies linearly according to mass of a load on the table while viscous friction is independent of load mass [58, 83]. In order to account for different loading conditions, (2.4) can be generalized into

$$\mathbf{M}'\ddot{\mathbf{x}}(t) + \mathbf{D}'\dot{\mathbf{x}}(t) + \mathbf{F}'_c \text{sgn}(\dot{\mathbf{x}}(t)) = \mathbf{K}_F \mathbf{i}(t), \quad (2.11)$$

with

$$\begin{aligned} \mathbf{M}' &= \mathbf{M} + m_{\text{load}} \mathbf{I}_{n(k)}, & \mathbf{D}' &= \mathbf{D}, & \mathbf{F}'_c &= \mathbf{F}_c (\mathbf{I}_{n(k)} + m_{\text{load}} \mathbf{M}_t^{-1}), \\ \mathbf{M}_t &= \text{diag} \{m_{t,k}\}, \end{aligned}$$

where m_{load} is the load mass on the table and $\mathbf{I}_{n(k)}$ is $n(k) \times n(k)$ identity matrix. $n(k)$ is the number of elements in set k . A load-dependent energy consumption model

$$\begin{aligned} P(t) &= \ddot{\mathbf{x}}(t)^\top \mathbf{C}'_1 \ddot{\mathbf{x}}(t) + \dot{\mathbf{x}}(t)^\top \mathbf{C}'_2 \dot{\mathbf{x}}(t) + \dot{\mathbf{x}}(t)^\top \mathbf{C}'_3 \text{sgn}(\dot{\mathbf{x}}(t)) + \text{tr}(\mathbf{C}'_4) + \ddot{\mathbf{x}}(t)^\top \mathbf{C}'_5 \text{sgn}(\dot{\mathbf{x}}(t)) \\ &\quad + \ddot{\mathbf{x}}(t)^\top \mathbf{C}'_6 \dot{\mathbf{x}}(t), \\ E &= \int_{t_0}^{t_f} P(t) dt, \end{aligned} \quad (2.12)$$

is derived by combining (2.11), (2.7) and (2.8), where

$$\begin{aligned} \mathbf{C}'_1 &= \mathbf{C}_1 (\mathbf{I}_{n(k)} + m_{\text{load}} \mathbf{M}^{-1})^2, & \mathbf{C}'_2 &= \mathbf{C}_2, \\ \mathbf{C}'_3 &= \mathbf{C}_3 (\mathbf{I}_{n(k)} + m_{\text{load}} \mathbf{M}_t^{-1}), & \mathbf{C}'_4 &= \mathbf{C}_4 (\mathbf{I}_{n(k)} + m_{\text{load}} \mathbf{M}_t^{-1})^2, \\ \mathbf{C}'_5 &= \mathbf{C}_5 (\mathbf{I}_{n(k)} + m_{\text{load}} \mathbf{M}_t^{-1}) (\mathbf{I}_{n(k)} + m_{\text{load}} \mathbf{M}^{-1}), & \mathbf{C}'_6 &= \mathbf{C}_6 (\mathbf{I}_{n(k)} + m_{\text{load}} \mathbf{M}^{-1}). \end{aligned} \quad (2.13)$$

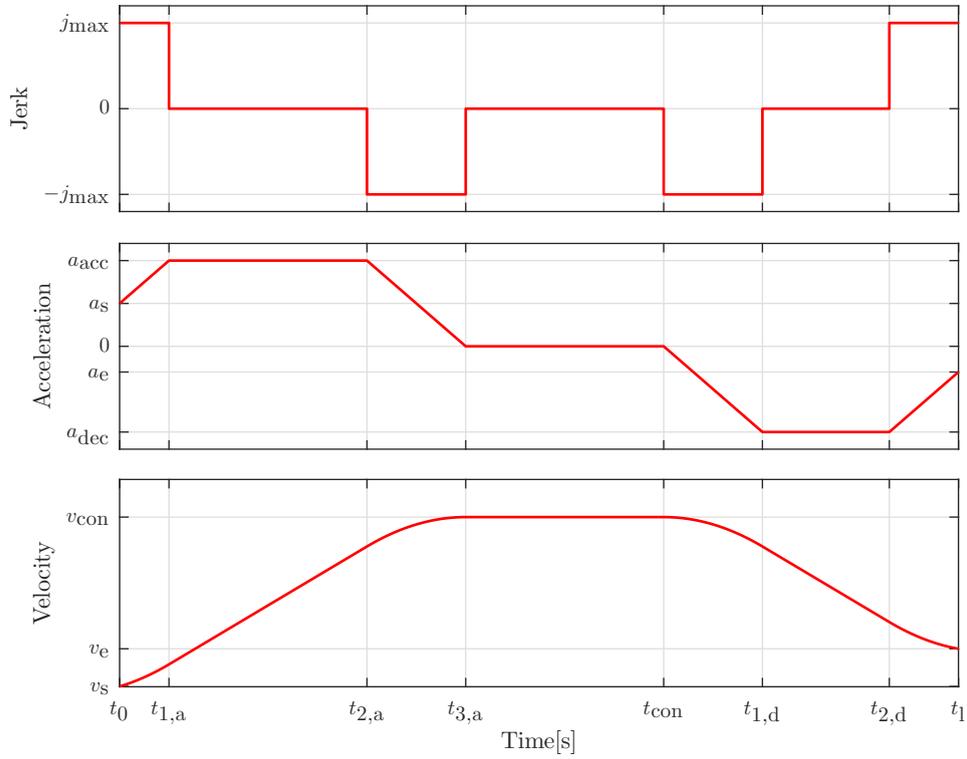


Fig. 2.5: An exemplary description of JLAPs.

2.3 Jerk-limited acceleration profiles

JLAPs are \mathbb{C}^2 continuous motion profiles that connect two points by providing acceleration, constant velocity and deceleration phases while obeying boundary conditions and restrictions on jerk, acceleration and velocity [6, 27, 66, 84]. An illustration of JLAP is shown in Fig. 2.5. The profile consists three transitions: an acceleration transition from t_0 to $t_{3,a}$, a constant velocity transition from $t_{3,a}$ to t_{con} and a deceleration transition from t_{con} to t_1 . A JLAP jerk

profile

$$\ddot{r}_k(t) = \begin{cases} j_{\max,k}, & t_{0,l} \leq t < t_{1,a}, \\ 0, & t_{1,a} \leq t < t_{2,a}, \\ -j_{\max,k}, & t_{2,a} \leq t < t_{3,a}, \\ 0, & t_{3,a} \leq t < t_{\text{con}} \\ -j_{\max,k}, & t_{\text{con}} \leq t < t_{1,d}, \\ 0, & t_{1,d} \leq t < t_{2,d}, \\ j_{\max,k}, & t_{2,d} \leq t < t_l \end{cases} \quad (2.14)$$

is defined, where $j_{\max,k}$ is the k^{th} axial jerk limit. By successive integration of (2.14), the acceleration and velocity profiles can be obtained. The time intervals

$$\begin{aligned} T_{1,a} &= t_{1,a} - t_0, & T_{1,d} &= t_{1,d} - t_{\text{con}}, \\ T_{2,a} &= t_{2,a} - t_{1,a}, & T_{2,d} &= t_{2,d} - t_{1,d}, \\ T_{3,a} &= t_{3,a} - t_{2,a}, & T_{3,d} &= t_l - t_{2,d}, \\ T_{\text{con}} &= t_{\text{con}} - t_{3,a}, \end{aligned} \quad (2.15)$$

determine respectively the acceleration, deceleration and velocity maxima

$$\begin{aligned} a_{\text{acc},k} &= a_{s,k} + j_{\max,k} T_{1,a}, \\ a_{\text{dec},k} &= -j_{\max,k} T_{1,d}, \\ v_{\text{con},k} &= v_{s,k} + a_{s,k} (T_{1,a} + T_{2,a} + T_{3,a}) + \frac{1}{2} j_{\max,k} (T_{1,a}^2 - T_{3,a}^2) + j_{\max,k} T_{1,a} (T_{2,a} + T_{3,a}), \end{aligned} \quad (2.16)$$

and terminal acceleration and velocity

$$\begin{aligned} a_{e,k} &= a_{\text{dec},k} + j_{\max,k} T_{3,d}, \\ v_{e,k} &= v_{\text{con},k} - \frac{1}{2} j_{\max,k} (T_{1,d}^2 - T_{3,d}^2) - j_{\max,k} T_{1,d} (T_{2,d} + T_{3,d}), \end{aligned} \quad (2.17)$$

where $a_{s,k}$ and $v_{s,k}$ are k^{th} axis starting acceleration and velocity, respectively.

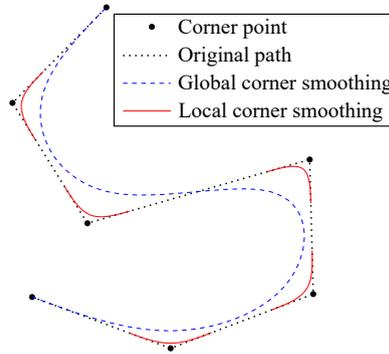


Fig. 2.6: A demonstration of the difference between global and local corner smoothing methods.

2.4 Corner smoothing methods

In the manufacturing industry, product geometry is described as a series of corner points interconnected by linear and arc segments (i.e., \mathbb{C}^0 continuous contours) using CAD/CAM systems. Motions along such segments require feed drive system to stop at each corner point. In order to reduce cycle time in manufacturing the product, corner smoothing methods have been proposed in the literature. These methods replace the corner points with fitted curves and provide smooth motion transitions from one segment to another. The shortest distance between each curve and corner point (i.e., a smoothing error) is constrained by a user-specified tolerance, thus defining a geometric constraint. The motion transitions are made within jerk, acceleration and velocity limits (i.e., kinematic constraints) of each feed drive system axis.

Based on the fitted curve's span, corner smoothing methods can be categorized into global and local corner smoothing methods. Global corner smoothing approaches fit a curve spanning across more than one corner point while local corner smoothing methods fit a curve confined at one corner point [81, 85, 98, 101–103] (Fig. 2.6). Global corner smoothing is typically used for short-segment contours; however, it more difficult to control the smoothing error compared to local corner smoothing [96]. Local corner smoothing approaches can be classified based on the simultaneity in considering the geometric and kinematic constraints [28, 72, 97]. Geometric local corner smoothing methods separately consider these constraints by firstly fitting a curve followed by scheduling a smooth motion transition along it. In the literature, several geometric local corner smoothing studies have been conducted: Yan et al. propose geometric local corner

smoothing using double cubic non-uniform rational B-spline (NURBS) for 5-axis CNC machine tool paths [105]. Yang and Yuen propose geometric local corner smoothing for improving machining efficiency by fitting 7th order polynomial splines [106]. A real-time \mathbb{G}^2 continuous geometric local corner smoothing with parameter synchronized B-splines of tool tip position and orientation paths is proposed for 5-axis CNC machine tools [41]. Pythagorean hodographs have been proposed for geometric local corner smoothing curve fitting [77, 96]. Curvature optimal quintic Bézier curves have been proposed for cycle time reduction [72]. Sun and Altintas propose \mathbb{G}^3 continuous geometric local corner smoothing using double Bézier curve fitting for 5-axis CNC machine tools [82].

Kinematic local corner smoothing approaches simultaneously consider geometric and kinematic constraints in blending velocity transitions from one segment to the next. Several kinematic local corner smoothing methods have been proposed in the literature: finite impulse response (FIR)-based kinematic local corner smoothing have been presented for reducing time and vibrations [73, 74, 86]. Xiao et al. propose \mathbb{G}^3 continuous 3D corner smoothing of line-line, line-arc, and arc-arc transitions using clothoids for improving machining quality and efficiency [100]. Huang et al. proposed clothoid-based kinematic local corner smoothing for improving contour accuracy [42]. Asymmetrical kinematic local corner smoothing with double constant-jerk cornering profiles have been illustrated in [98]. Tajima and Sencer propose KCS for near cycle time optimality [84].

2.4.1 Kinematic corner smoothing

KCS is a JLAP-based 2D kinematic local corner smoothing approach that analytically computes cornering velocity and acceleration by exploiting kinematic constraints to achieve near time optimal motion profiles while maintaining geometric constraints [84]. The resulting cornering trajectory is then blended with the interconnecting segments at an order of \mathbb{C}^2 continuity. Fig. 2.7 shows a KCS curve starting at \mathbf{p}_s and ending at \mathbf{p}_e . The smoothing error ε is defined from the Euclidean distance between the curve's mid-point \mathbf{p}_{mid} and the original corner point \mathbf{p}_c . In [84], Tajima and Sencer describe two KCS methods: KCSIA and kinematic corner smoothing with uninterrupted acceleration (KCSUA) which are described below.

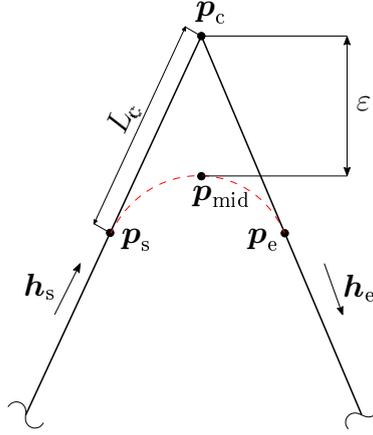


Fig. 2.7: Geometric description of a KCS curve.

2.4.1.1 Kinematic corner smoothing with interrupted acceleration

KCSIA sets zero tangential acceleration at the start and end of the cornering profile with the same tangential velocity magnitude V_c . Fig. 2.8 illustrates a KCSIA cornering profile, where the motion is subdivided into positive, zero and negative jerk sections as follows:

$$\ddot{r}_k(t) = \begin{cases} j_{c,k}, & t_{0,c} \leq t < t_{1,c}, \\ 0, & t_{1,c} \leq t < t_{2,c}, \\ -j_{c,k}, & t_{2,c} \leq t < t_c, \end{cases} \quad (2.18)$$

is defined with

$$\begin{aligned} j_{c,k} &= \frac{v_{e,k} - v_{s,k}}{T_{1,c}(T_{1,c} + T_{2,c})}, & T_{1,c} &= t_{1,c} - t_0 \\ v_{s,k} &= V_c h_{s,k}, & &= t_c - t_{2,c}, \\ v_{e,k} &= V_c h_{e,k}, & T_{2,c} &= t_{2,c} - t_{1,c}. \\ a_{c,k} &= j_{c,k} T_{1,c}, \end{aligned} \quad (2.19)$$

$j_{c,k}$ is the cornering jerk. $h_{s,k}$ and $h_{e,k}$ are the k^{th} axis unit vector components of tangential velocity at the start and end of the cornering profile, respectively. $v_{s,k}$ and $v_{e,k}$ are respectively the k^{th} axis velocity components at the cornering start and end. The start, middle and end

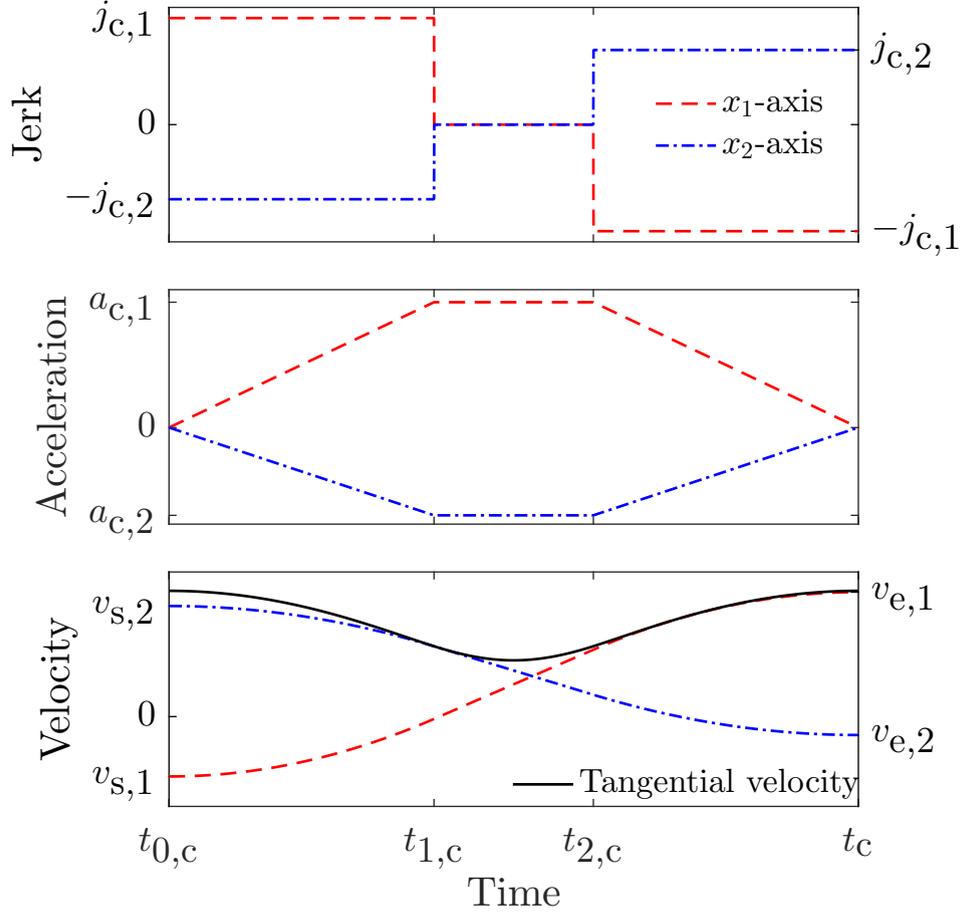


Fig. 2.8: Exemplary jerk, acceleration and velocity profiles generated by KCSIA.

points of the profile

$$\begin{aligned}
 p_{s,k} &= p_{c,k} - L_c h_{s,k}, \\
 p_{\text{mid},k} &= p_{s,k} + v_{s,c,k} \left[T_{1,c} + \frac{1}{2} T_{2,c} \right] + j_{c,k} \left[\frac{1}{2} T_{1,c} \left(\frac{T_{2,c}}{2} \right)^2 \right] + j_{c,k} \left[\frac{1}{6} T_{1,c}^3 + \frac{1}{2} T_{1,c}^2 \left(\frac{T_{2,c}}{2} \right) \right] \quad (2.20) \\
 p_{e,k} &= p_{c,k} + L_c h_{e,k}
 \end{aligned}$$

are described with a cornering Euclidean length

$$L_c = \frac{1}{2} V_c (2T_{1,c} + T_{2,c}). \quad (2.21)$$

From (2.18) and (2.20), the KCSIA motion transition can be defined using V_c , $T_{1,c}$ and $T_{2,c}$, where

$$\varepsilon = \frac{V_c \{T_{1,c}^2 + 3(T_{1,c} + T_{2,c})^2\}}{24(T_{1,c} + T_{2,c})} \|\mathbf{h}_e - \mathbf{h}_s\|_2. \quad (2.22)$$

is limited by a user-specified tolerance. KCSIA proceeds to solve the optimization problem

$$\begin{aligned} \max_{\boldsymbol{\mu}_c} V_c \\ \boldsymbol{\mu}_c = [V_c, T_{1,c}, T_{2,c}] \end{aligned} \quad (2.23)$$

subject to kinematic and geometric constraints. The resulting cornering profile is then interconnected with time-optimal JLAP generated linear motions.

2.4.1.2 Kinematic corner smoothing with uninterrupted acceleration

KCSUA differs from KCSIA by setting a non-zero tangential acceleration A_c at the start and end of the cornering motion. As shown in Fig: 2.9, a KCSUA jerk profile

$$\ddot{r}_k(t) = j_{c,k}, \quad t_0 \leq t < t_c, \quad (2.24)$$

is defined with

$$\begin{aligned} j_{c,k} &= \frac{a_{e,k} - a_{s,k}}{T_{1,c}}, & a_{s,k} &= -A_c h_{s,k}, \\ T_{1,c} &= t_c - t_0, & a_{e,k} &= A_c h_{e,k}, \end{aligned}$$

where $a_{s,k}$ and $a_{e,k}$ are the k^{th} axis tangential acceleration components at the start and end of the cornering profile, respectively. The start, middle and end points of the profile are derived as

$$\begin{aligned} \mathbf{p}_s &= \mathbf{p}_c - L_c \mathbf{h}_s \\ \mathbf{p}_{\text{mid}} &= \mathbf{p}_s + \mathbf{v}_s \left(\frac{T_{1,c}}{2} \right) + \frac{1}{2} \mathbf{a}_s \left(\frac{T_{1,c}}{2} \right)^2 + \frac{1}{6} \mathbf{j}_c \left(\frac{T_{1,c}}{2} \right)^3 \\ \mathbf{p}_e &= \mathbf{p}_c + L_c \mathbf{h}_e. \end{aligned} \quad (2.25)$$

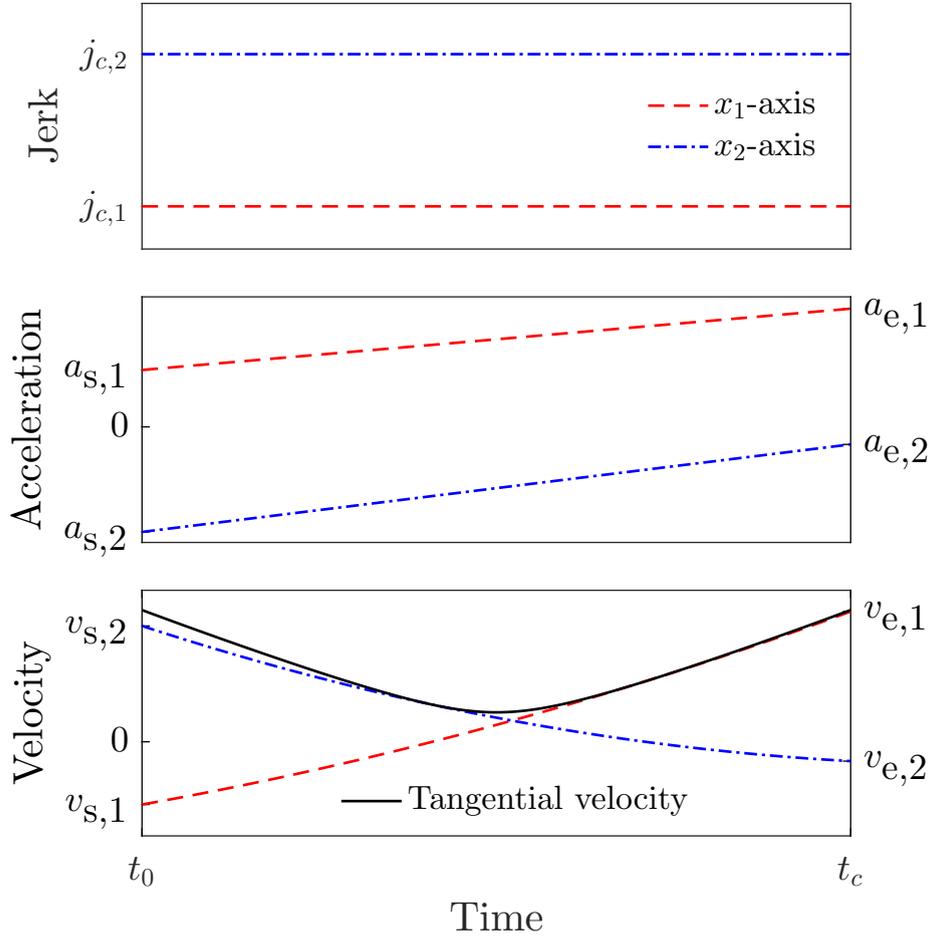


Fig. 2.9: Exemplary jerk, acceleration and velocity profiles generated by KCSUA.

with $L_c = \frac{1}{6}A_c T_{1,c}^2$ and $V_c = \frac{1}{2}A_c T_{1,c}$. From (2.24) and (2.25), the KCSUA profile can be defined using A_c and $T_{1,c}$, where

$$\varepsilon = \frac{A_c T_{1,c}^2}{48} \|\mathbf{h}_e - \mathbf{h}_s\|_2. \quad (2.26)$$

is limited by a user-specified tolerance. KCSUA solves the optimization problem

$$\begin{aligned} \max_{\boldsymbol{\mu}_c} V_c \\ \boldsymbol{\mu}_c = [V_c, T_{1,c}] \end{aligned} \quad (2.27)$$

subject to kinematic and geometric constraints followed by interconnecting the resulting cornering profile with time-optimal JLAP generated linear motions.

2.5 Multi-objective optimization methods

Problems requiring the optimization of more than one objective are termed as multi-objective optimization problems (MOOPs). A MOOP with m_o objectives can be represented as

$$\min_{\boldsymbol{\mu} \in \Lambda \subseteq \mathbb{R}^{n_{\text{var}}}} \mathbf{J}(\boldsymbol{\mu}) = [J_1(\boldsymbol{\mu}), J_2(\boldsymbol{\mu}), \dots, J_{m_o}(\boldsymbol{\mu})]^\top, \quad m_o \geq 2, \quad (2.28)$$

where

$$\Lambda = \{\boldsymbol{\mu} : \mathbf{g}(\boldsymbol{\mu}) = \mathbf{0}, \quad \mathbf{q}(\boldsymbol{\mu}) \leq \mathbf{0}, \quad \boldsymbol{\mu}_{\text{lb}} \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}_{\text{ub}}\}, \quad (2.29)$$

with $\mathbf{J} : \mathbb{R}^{n_{\text{var}}} \mapsto \mathbb{R}^{m_o}$, $\mathbf{g} : \mathbb{R}^{n_{\text{var}}} \mapsto \mathbb{R}^{n_g}$ and $\mathbf{q} : \mathbb{R}^{n_{\text{var}}} \mapsto \mathbb{R}^{n_q}$, where \mathbf{g} and \mathbf{q} are respectively the equality and inequality constraint vectors. $\boldsymbol{\mu}_{\text{lb}}$ and $\boldsymbol{\mu}_{\text{ub}}$ are the lower and upper bounding vectors of the optimization variable $\boldsymbol{\mu}$, respectively. n_{var} , n_g and n_q are accordingly the number of variables, equality and inequality constraints. A candidate solution in a feasible region of decision space Λ can be mapped to an objective space Ω as shown in Fig. 2.10, where a set of non-dominated optimal solutions form a Pareto frontier Ψ . $\boldsymbol{\mu}_a$ dominates $\boldsymbol{\mu}_b$ if $J_{a,m} \leq J_{b,m} \forall m \in \{1, 2, \dots, m_o\}$ and $\exists m \in \{1, 2, \dots, m_o\} : J_{a,m} < J_{b,m}$. The utopia point \mathbf{O} is an unattainable point in the objective space where all m objectives are minimized. Convex hull of individual minima (CHIM) is a hyperplane connecting solutions that optimize a single objective (i.e., anchor points $\boldsymbol{\psi}_m$). In 2D case, CHIM is a line segment as shown in Fig. 2.10.

A concave feasible region of objective space can have Pareto frontiers with global and local-optimal regions. Messac et al. describe the concept of local and global Pareto optimality in [55]. A candidate solution $\hat{\boldsymbol{\psi}}$ is globally Pareto-optimal if there does not exist another solution $\boldsymbol{\psi} \in \Psi$ such that

$$\forall m \in \{1, 2, \dots, m_o\} : \psi_m \leq \hat{\psi}_m, \quad \text{and} \quad \exists m \in \{1, 2, \dots, m_o\} : \psi_m < \hat{\psi}_m. \quad (2.30)$$

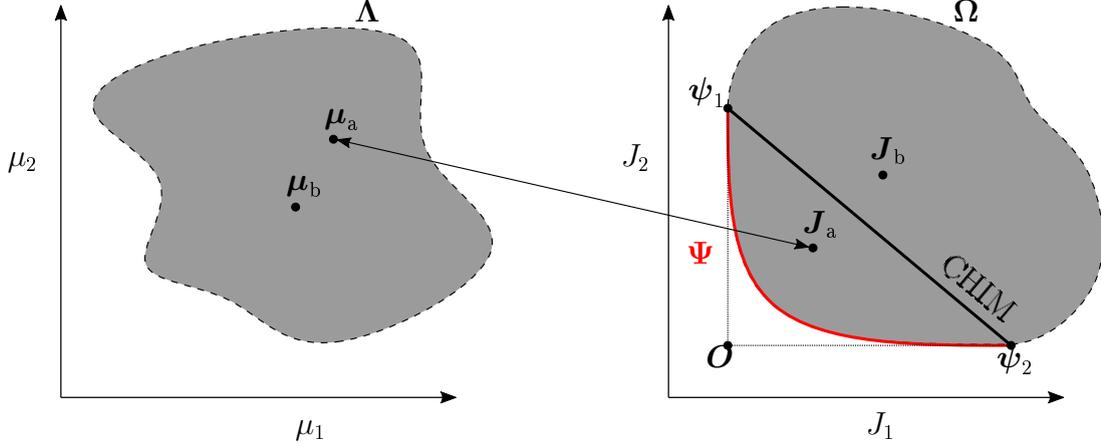


Fig. 2.10: An illustration of mapping a candidate solution in a feasible region of decision space to/from an objective space. The shaded area marks the feasible region. The Pareto frontier is mark as a solid red curve. \mathbf{O} marks the utopia point.

On the other hand, for local Pareto optimality, the condition holds only in the neighborhood of $\hat{\boldsymbol{\psi}}$. Fig. 2.11 shows a demonstration of global and local Pareto frontiers of a concave feasible region. Segments $\boldsymbol{\psi}_1\boldsymbol{\psi}_a$ and $\boldsymbol{\psi}_c\boldsymbol{\psi}_2$ represent a global Pareto frontier while segment $\boldsymbol{\psi}_b\boldsymbol{\psi}_c$ represents a local Pareto frontier. Segment $\boldsymbol{\psi}_a\boldsymbol{\psi}_b$ is a non-Pareto region.

In the literature, there are two classes of multi-objective optimization methods for generating approximate Pareto frontiers: vectorization and scalarization methods [36, 52]. Vectorization methods are evolutionary algorithms that generate, in one simulation run, multiple optimal solutions distributed across the Pareto frontier [22, 23, 80]. Although these methods produce globally optimal solutions, multiple objective function evaluations for each candidate solution at every iteration may result in high computation cost[52]. The difficulty in considering constraints other than bounding ones also limits the application of these methods [36]. Scalarization methods generate optimal solutions by recursively converting a MOOP into a single-objective optimization problem (SOOP) resulting in one locally optimal solution per recursion. In order to convert an objective vector \mathbf{J} into a scalar, these methods normalize each objective as follows

$$\tilde{J}_m(\boldsymbol{\mu}) = \frac{J_m(\boldsymbol{\mu}) - J_{m,\min}}{J_{m,\max} - J_{m,\min}}, \quad m \in \{1, 2, \dots, m_o\}, \quad (2.31)$$

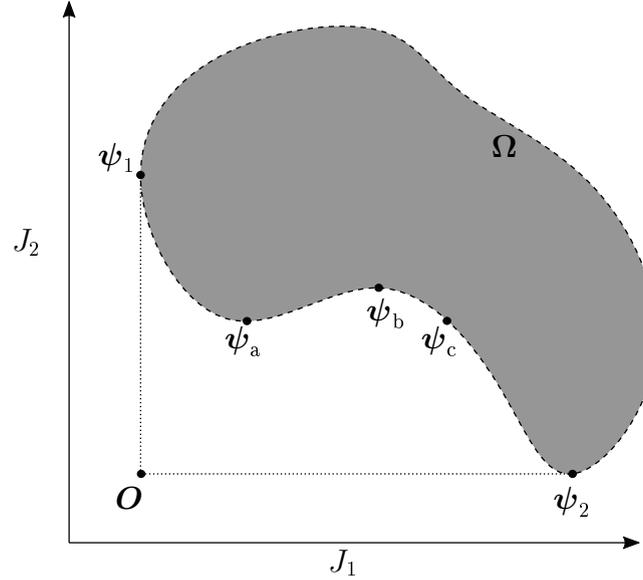


Fig. 2.11: An illustration of global and local Pareto frontiers of a concave feasible region Ω .

with

$$J_{m,\min} = J_m(\mu_m), \quad \mu_m = \arg \min_{\mu} J_m(\mu),$$

$$J_{m,\max} = \max \{J_m(\mu_1), J_m(\mu_2), \dots, J_m(\mu_{m_o})\}.$$

The weighted sum approach is a commonly used scalarization method. (2.28) is reformulated as

$$\min_{\mu \in \Lambda \subseteq \mathbb{R}^{n_{\text{var}}}} J_{\text{ws}} = \sum_{m=1}^{m_o} \zeta_m \tilde{J}_m(\mu), \quad \sum_{m=1}^{m_o} \zeta_m = 1, \quad \zeta_m \geq 0, \quad (2.32)$$

where ζ_m is the weighting factor of objective J_m . The weighted sum method assumes convexity of the feasible region in objective space. Thus, it cannot generate solutions in concave Pareto regions [20]. To counter this limitation, the ϵ -constraint method [87] has been proposed. This approach converts a MOOP into a SOOP by minimizing one objective while the remaining objectives are used as bounding constraints in the objective space (Fig. 2.12). The ϵ -constraint

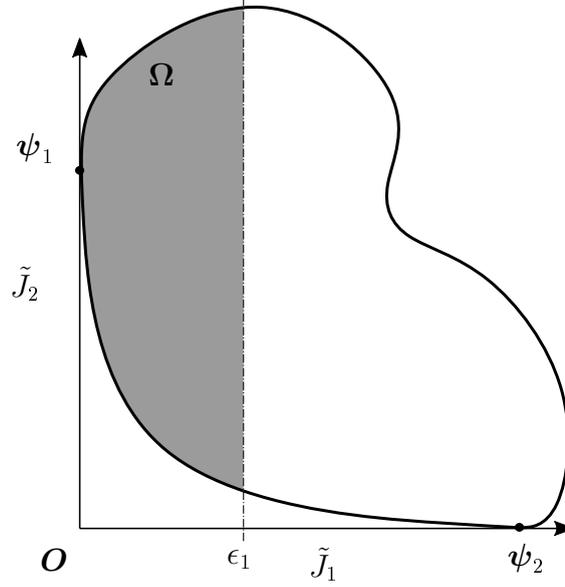


Fig. 2.12: A demonstration of the ϵ -constraint method.

method is represented as

$$\min_{\mu \in \Lambda_{\subseteq \mathbb{R}^n \text{ var}}} \tilde{J}_{m_\epsilon}(\boldsymbol{\mu}), \quad (2.33)$$

subject to

$$\tilde{J}_m(\boldsymbol{\mu}) - \epsilon_m \leq 0, \quad m \in \{1, 2, \dots, m_o\}, \quad m \neq m_\epsilon, \quad (2.34)$$

where each solution corresponds to a set of ϵ_m . This approach suffers a drawback in Pareto point distribution since uniformly distributing ϵ_m does not correlate to a uniform distribution of optimal solutions on the Pareto frontier. In order to mitigate this drawback, normal boundary intersection (NBI) [21], NNC [55, 69] are proposed methods in the literature. The NBI method constructs uniformly distributed quasi-normal lines to CHIM and maximizes the distances from CHIM to the Pareto frontier along these lines to generate a uniform distribution Pareto-optimal solutions (Fig. 2.13). The NBI approach reformulates (2.28) as follows:

$$\max_{\mu \in \Lambda_{\subseteq \mathbb{R}^n \text{ var}}} \|\tilde{\mathbf{J}}(\boldsymbol{\mu}) - \boldsymbol{\rho}\|_2, \quad (2.35)$$

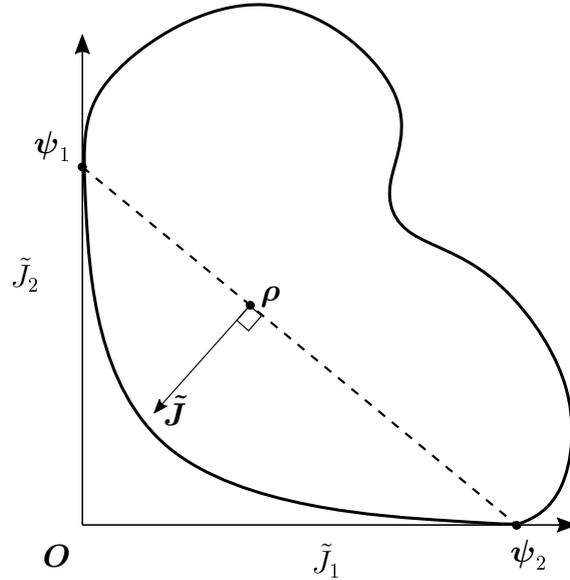


Fig. 2.13: A demonstration of the NBI method.

subject to

$$[\psi_{m+1} - \psi_m]^\top [\tilde{\mathbf{J}}(\boldsymbol{\mu}) - \boldsymbol{\rho}] = 0, \quad m \in \{1, 2, \dots, m_o - 1\}, \quad (2.36)$$

with a CHIM point

$$\boldsymbol{\rho} = \sum_{m=1}^{m_o} \zeta_m \psi_m, \quad \sum_{m=1}^{m_o} \zeta_m = 1, \quad \zeta_m \geq 0. \quad (2.37)$$

The NBI method has a drawback of producing non-Pareto optimal points in cases with concave feasible regions in the objective space. The NNC method counters this limitation by minimizing one objective and replacing (2.36) with inequality constraints incorporating the remaining objectives. Thus, the NNC method provides a reduced feasible region for finding Pareto-optimal solutions as shown in Fig. 2.14. The NNC method reformulates (2.28) as follows:

$$\min_{\boldsymbol{\mu} \in \Lambda \subseteq \mathbb{R}^{p_{\text{var}}}} \tilde{J}_{m_N}(\boldsymbol{\mu}), \quad (2.38)$$

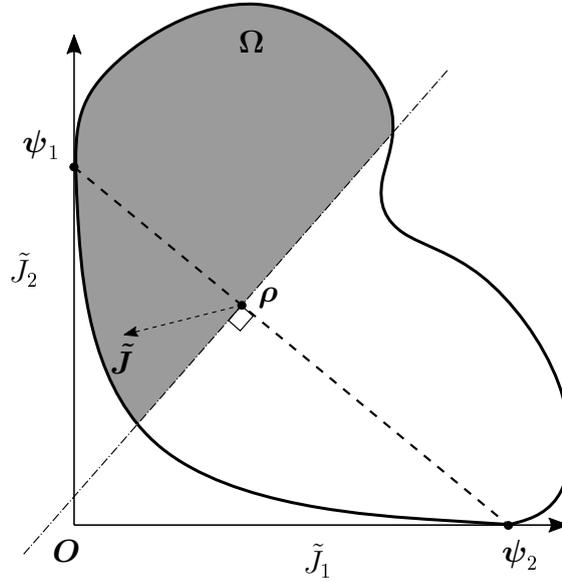


Fig. 2.14: A demonstration of the NNC method.

subject to

$$[\psi_{m_N} - \psi_m]^\top [\tilde{\mathbf{J}}(\mu) - \rho] \leq 0, \quad m \in \{1, 2, \dots, m_o\}, \quad m \neq m_N. \quad (2.39)$$

with (2.37). Depending on the choice of m_N , the NNC approach may produce non-Pareto optimal solutions. In order to detect these undesirable solutions irrespective of the objective choice, Logist and Van Impe propose a removal criterion for such solutions [53]. Candidate solutions in non-Pareto optimal regions deactivate (2.39), hence making the corresponding Lagrange multipliers ν become zero. Such a candidate solution is not added into Ψ if not all the first $m_o - 1$ elements of

$$\hat{\nu}_m = \frac{1}{m_o - 1} \mathbf{E}^{-1} \mathbf{P}^{m_o - m} \mathbf{E} \nu_{\text{aug}} \quad (2.40)$$

$$m \in \{m_o, m_o - 1, \dots, 1\}$$

are positive, where $\mathbf{E} = \mathbf{1}_{m_o} - \mathbf{I}_{m_o}$. \mathbf{P} , $\mathbf{1}_{m_o}$ and \mathbf{I}_{m_o} are $m_o \times m_o$ permutation, all-ones and identity matrices, respectively.

$$\boldsymbol{\nu}_{\text{aug}} = \left[\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_{m_o-1}, \sum_{m=1}^{m_o-1} \boldsymbol{\nu}_m \right]^\top \quad (2.41)$$

is an augmented vector of Lagrange multipliers of (2.39).

2.6 Divide and conquer algorithm

Multi-objective optimization methods generate approximations of a Pareto frontier with optimal solutions spread across significant (i.e., knee) and insignificant (i.e., plateau) regions of the frontier. Decision-makers are then required to use a filter to retain Pareto regions with significant trade-offs. Mattson et al. propose a smart filter for trimming insignificant Pareto-optimal solutions [54]. Filters that rank solutions based on efficiency have been studied in [4, 19]. Farina and Amato propose the reduction of insignificant Pareto-optimal solutions using a fuzzy optimality-based dominance concept [32]. An evolutionary algorithm for reducing solutions on plateau regions is proposed in [11]. The process of generating Pareto-optimal solutions followed by insignificant solution filtering is computationally expensive. Hashem et al. propose the divide and conquer algorithm for generating Pareto frontier approximations with adaptive resolution [36].

The divide and conquer algorithm recursively uses the NBI or NNC method to explore the CHIM for regions with significant trade-off. The Pareto set Ψ is initialized with the insertion of m_o anchor points. A candidate solution $\boldsymbol{\psi}$ is obtained by taking a centroid ρ of CHIM points corresponding to a cluster of m_o neighboring Pareto-optimal points in Ψ and solving an SOOP. $\boldsymbol{\psi}$ is added into Ψ if it is significant compared to the cluster members. For a cluster having points $\boldsymbol{\psi}_{p_1}, \boldsymbol{\psi}_{p_2}, \dots, \boldsymbol{\psi}_{p_{m_o}}$, $\boldsymbol{\psi}$ is significant if

$$\min(\delta_{p_1}, \delta_{p_2}, \dots, \delta_{p_{m_o}}) \geq \delta_{\min} \quad (2.42)$$

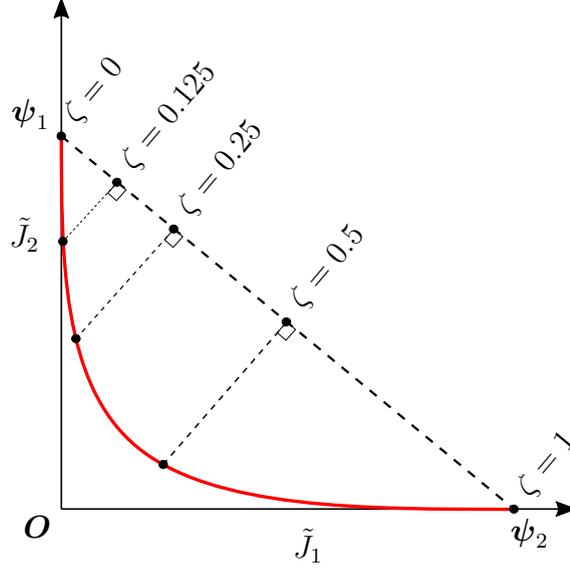


Fig. 2.15: Illustration of the divide and conquer algorithm for a two-objective case.

with

$$\delta_{p_m} = \min\left(|\psi_{p_m,1} - \psi_1|, |\psi_{p_m,2} - \psi_2|, \dots, |\psi_{p_m,m_o} - \psi_{m_o}|\right), \quad m \in \{1, 2, \dots, m_o\},$$

$$\boldsymbol{\psi} = [\psi_1, \psi_2, \dots, \psi_{m_o}]^\top,$$

$$\boldsymbol{\psi}_{p_m} = [\psi_{p_m,1}, \psi_{p_m,2}, \dots, \psi_{p_m,m_o}]^\top,$$

where δ_{\min} is a user-specified minimum trade-off level. Subsequently, a new cluster, having the previous $\boldsymbol{\psi}$ and $m_o - 1$ cluster members, is explored until an insignificant $\boldsymbol{\psi}$ obtained, where it is not added into $\boldsymbol{\Psi}$ and the next unexplored cluster in $\boldsymbol{\Psi}$ is explored. Fig. 2.15 demonstrates the working principle of the divide and conquer algorithm for $m_o = 2$. The divide and conquer algorithm pseudo code is shown in Algorithm 1.

2.7 Global Pareto-optimal filter

The removal (2.40) and significance (2.42) criteria cannot detect local Pareto regions of $\boldsymbol{\Psi}$. Hence a global Pareto-optimal filter [55] is implemented in this thesis to retain only global

Algorithm 1 The divide and conquer algorithm

Input: δ_{\min}
Output: Ψ ▷ A set containing all Pareto points

- 1: **procedure** DIVIDE AND CONQUER
- 2:
- 3: Calculate *anchor points*
- 4: $\Psi \leftarrow \{\}$ ▷ Initialize the Pareto set
- 5: $\Psi \leftarrow \{\Psi, \text{anchor points}\}$ ▷ Add anchor points to Pareto set
- 6: *current cluster* \leftarrow *anchor points*
- 7:
- 8: **while** unexplored clusters exist **do**
- 9: Calculate centroid ρ at *current cluster* ▷ a point on CHIM
- 10: $\psi \leftarrow$ Solve SOOP at ρ
- 11:
- 12: $b_s \leftarrow$ significanceCriterion(*current cluster*, ψ , δ_{\min}) ▷ See (2.42)
- 13:
- 14: **if** b_s **then**
- 15: $\Psi \leftarrow \{\Psi, \psi\}$
- 16: *current cluster* \leftarrow ψ and $m_o - 1$ *current cluster* members
- 17: **else**
- 18: *current cluster* \leftarrow next unexplored cluster
- 19: **end if**
- 20:
- 21: **end while**
- 22:
- 23: Return Ψ
- 24: **end procedure**

Pareto points once the divide and conquer algorithm is terminated. Algorithm 2 illustrates the filter's pseudo code.

Algorithm 2 The global Pareto-optimal filter algorithm

Input: Ψ ▷ A set containing global & local Pareto-optimal points
Output: Ψ_g ▷ A set containing only global Pareto-optimal points

```

1: procedure GLOBALPARETOFILTER
2:
3:    $\Psi_g \leftarrow \{\}$  ▷ Initialize the Pareto set
4:
5:   for  $i = 0;$   $i < n(\Psi);$   $i++$  do ▷ loop through all elements of  $\Psi$ 
6:
7:      $\psi \leftarrow \Psi(i)$ 
8:
9:      $b_g \leftarrow \text{globalOptimalityCriterion}(\psi, \Psi)$  ▷ See (2.30)
10:
11:     if  $b_s$  then
12:        $\Psi_g \leftarrow \{\Psi_g, \psi\}$ 
13:     end if
14:
15:   end for
16:
17:   Return  $\Psi_g$ 
18: end procedure

```

Chapter 3

Pareto Optimization of Energy and Tolerance in Motion Trajectory Generation

3.1 Introduction

In the manufacturing industry, CNC machine tools are typically used to produce components, where accuracy, high production rates and low energy costs are key requirements. They are actuated by feed drive systems [3], and studies related to improving accuracy [12, 51], increasing production rates [25, 72, 79] and minimizing energy consumption [37, 91, 92] have been conducted.

Accuracy, cycle time and energy consumption compose a MOOP that has an infinite set of Pareto optimal solutions. Studies have been conducted in formulating methods for representing Pareto fronts [20, 21, 36, 55]. In the literature, local corner smoothing algorithms have been studied for the purpose of reducing cycle time, where cornering velocities are maximized and cornering errors are bounded by user-specified accuracy constraint [30, 41, 72, 84, 105, 106]. In economic lot scheduling of supply chains, different cycle times (i.e., optimal or otherwise) can be selected to reduce overall production costs [56, 88]. These corner smoothing algorithms are limited to generating cornering trajectories under time-optimal conditions.

In the previous work [64], a trade-off between time and energy consumption in generating corner smoothed trajectories for piece-wise linear contours was studied. This method is extended in this work, where energy consumption is optimized in a fixed cycle time situation. Thus, energy consumption and cornering error are used as objectives for the optimization problem. Linear segments are defined using a JLAP [6, 27, 44]. Smoothed corner trajectories are described by KCSIA [84]. The NNC [55] and SQP [9, 33, 62] are utilized to formulate and solve the optimization problem, respectively. Pareto optimal solutions are represented using the divide and conquer algorithm [36], where the best trade-off solution is selected.

In summary, the contributions of this work are as follows:

- A fixed cycle time corner smoothing method is proposed.
- The method offers Pareto-optimal solutions for energy consumption and cornering error.

The rest of this chapter is organized as follows: Trajectory representation is shown in Section 3.2, where JLAP and KCSIA trajectories are described in Sections 3.2.1 and 3.2.2, respectively. Trajectory optimization is presented in Section 3.3, followed by simulation results shown in Section 3.4. Conclusions of this work are presented in Section 3.5.

3.2 Trajectory representation

This section presents the description of trajectories for geometries with piecewise linear segments and smoothed corners. Linear segment trajectories are defined by JLAP while smoothed cornering paths are described using KCSIA.

3.2.1 Jerk-limited acceleration profiles

JLAP is generally composed of three phases with jerk restrictions, namely, acceleration, constant velocity and deceleration phases as follows:

$$\begin{aligned}
 \ddot{\mathbf{x}}_{a,k}(t) &= \begin{cases} j_{\max,k}, & t_0 \leq t < t_{1,a}, \\ 0, & t_{1,a} \leq t < t_{2,a}, \\ -j_{\max,k}, & t_{2,a} \leq t < t_{3,a}, \end{cases} \\
 \ddot{\mathbf{x}}_{\text{con},k}(t) &= 0, \quad t_{3,a} \leq t < t_{\text{con}}, \\
 \ddot{\mathbf{x}}_{d,k}(t) &= \begin{cases} -j_{\max,k}, & t_{\text{con}} \leq t < t_{1,d}, \\ 0, & t_{1,d} \leq t < t_{2,d}, \\ j_{\max,k}, & t_{2,d} \leq t < t_1, \end{cases}
 \end{aligned} \tag{3.1}$$

where the jerk profiles for the acceleration, constant velocity and deceleration phases are given by $\ddot{\mathbf{x}}_{a,k}$, $\ddot{\mathbf{x}}_{\text{con},k}$, and $\ddot{\mathbf{x}}_{d,k}$, respectively. $j_{\max,k}$ is the k^{th} axial jerk limit. The acceleration, velocity and displacement profiles can be obtained by successively integrating (3.1). The time intervals

$$\begin{aligned}
 T_{1,a} &= t_{1,a} - t_{0,l}, & T_{1,d} &= t_{1,d} - t_{\text{con}}, \\
 T_{2,a} &= t_{2,a} - t_{1,a}, & T_{2,d} &= t_{2,d} - t_{1,d}, \\
 T_{3,a} &= t_{3,a} - t_{2,a}, & T_{3,d} &= t_1 - t_{2,d}, \\
 T_{\text{con}} &= t_{\text{con}} - t_{3,a},
 \end{aligned} \tag{3.2}$$

are determined such that trajectory satisfies the acceleration, velocity and spatial constraints. Hence, the linear segment trajectory can be defined by a variable vector

$$\boldsymbol{\mu}_1 = [T_{1,a}, T_{2,a}, T_{3,a}, T_{\text{con}}, T_{1,d}, T_{2,d}, T_{3,d}]^T, \tag{3.3}$$

where the total duration is

$$T_1(\boldsymbol{\mu}_1) = T_{1,a} + T_{2,a} + T_{3,a} + T_{\text{con}} + T_{1,d} + T_{2,d} + T_{3,d}. \tag{3.4}$$

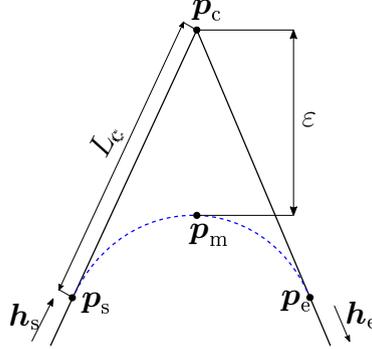


Fig. 3.1: Geometric description of the KCSIA approach.

According to (Section 2.2), the energy consumed in traversing the trajectory is obtained as

$$E_1(\mu_1) = \int_{t_0}^{t_1} P(t) dt. \quad (3.5)$$

3.2.2 Kinematic corner smoothing with interrupted acceleration

KCSIA is a method that generates 2D smoothed corner trajectories by analytically evaluating the cornering velocity and duration while setting a zero path acceleration, at the start and end of the cornering motion [84] as shown in Fig. 3.1. A predetermined cornering tolerance $0 < \varepsilon \leq \varepsilon_{ub}$ and feed drive kinematic limits are utilized as constraints to plan a jerk limited cornering trajectory. \mathbb{C}^2 motion continuity is guaranteed by equating the accelerations and velocities at the nodes (i.e. points connecting the smoothed corner with the adjacent linear segments). In Fig. 3.1, \mathbf{p}_s and \mathbf{p}_e are the nodes and the maximum cornering error ε is located at the trajectory mid-point \mathbf{p}_m . The original corner is located at \mathbf{p}_c . The jerk profile of the cornering motion is given by

$$\ddot{\mathbf{x}}_k(t) = \begin{cases} j_{c,k}, & t_0 \leq t < t_{1,c}, \\ 0, & t_{1,c} \leq t < t_{2,c}, \\ -j_{c,k}, & t_{2,c} \leq t < t_c, \end{cases} \quad (3.6)$$

with

$$\begin{aligned}
 j_{c,k} &= \frac{v_{e,c,k} - v_{s,c,k}}{T_{1,c}(T_{1,c} + T_{2,c})}, & v_{s,c,k} &= V_c h_{s,k}, \\
 T_{1,c} &= t_{1,c} - t_0 = t_c - t_{2,c}, & v_{e,c,k} &= V_c h_{e,k}, \\
 T_{2,c} &= t_{2,c} - t_{1,c}, & &
 \end{aligned} \tag{3.7}$$

where $j_{c,k}$ is the cornering jerk. V_c is the path velocity magnitude at the nodes. $v_{s,c,k}$ and $h_{s,k}$ are the starting point path velocity and its unit vector component, respectively. Accordingly, $v_{e,c,k}$ and $h_{e,k}$ are the end point path velocity and its unit vector component.

The total displacement in each axis is

$$\Delta s_{c,k} = \left(v_{s,c,k} + \frac{1}{2} j_{c,k} T_{1,c} (T_{1,c} + T_{2,c}) \right) (2T_{1,c} + T_{2,c}), \tag{3.8}$$

which is derived by successive integration of (3.6), is used for expressing the Euclidean length

$$L_c = \frac{\Delta s_k}{(h_{s,k} + h_{e,k})}. \tag{3.9}$$

From (3.9), expressions for \mathbf{p}_s , \mathbf{p}_m and \mathbf{p}_e are obtained as follows:

$$\begin{aligned}
 \mathbf{p}_s &= \mathbf{p}_c - L_c \mathbf{h}_s, \\
 \mathbf{p}_m &= \mathbf{p}_s + \mathbf{v}_{s,c} \left(T_{1,c} + \frac{1}{2} T_{2,c} \right) + \mathbf{j}_c \left\{ \frac{1}{2} T_{1,c} \left(\frac{T_{2,c}}{2} \right)^2 \right\} + \mathbf{j}_c \left\{ \frac{1}{6} T_{1,c}^3 + \frac{1}{2} T_{1,c}^2 \left(\frac{T_{2,c}}{2} \right) \right\}, \\
 \mathbf{p}_e &= \mathbf{p}_c + L_c \mathbf{h}_e,
 \end{aligned} \tag{3.10}$$

whereby a cornering constraint

$$\|\mathbf{p}_c - \mathbf{p}_m\|_2 - \varepsilon = 0, \tag{3.11}$$

is introduced. The smoothed corner trajectory can be defined by a variable vector

$$\boldsymbol{\mu}_c = [V_c, T_{1,c}, T_{2,c}]^T, \tag{3.12}$$

with the total cornering duration

$$T_c(\boldsymbol{\mu}_c) = 2T_{1,c} + T_{2,c}. \quad (3.13)$$

In a similar manner to the previous section, the energy required for the cornering motion is described as

$$E_c(\boldsymbol{\mu}_c) = \int_{t_0}^{t_c} P(t) dt. \quad (3.14)$$

3.3 Trajectory optimization

3.3.1 Problem formulation

This section presents a BOOP using the energy consumption model, JLAP and KCSIA described in sections 2.2, 3.2.1 and 3.2.2, respectively. The methods in [55] and [64] are extended to formulate the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{\mu}} \{ \varepsilon, E_{\text{tot}}(\boldsymbol{\mu}) \}, \\ \boldsymbol{\mu} = [\varepsilon, \boldsymbol{\mu}_{1,1}, \boldsymbol{\mu}_{c,1}, \boldsymbol{\mu}_{1,2}, \boldsymbol{\mu}_{c,2}, \dots, \boldsymbol{\mu}_{1,n_1}, \boldsymbol{\mu}_{c,n_c}]^T, \end{aligned} \quad (3.15)$$

subject to

$$g_r(\boldsymbol{\mu}) = 0, \quad r = \{1, 2, \dots, n_g\}, \quad (3.16)$$

$$q_y(\boldsymbol{\mu}) \leq 0, \quad y = \{1, 2, \dots, n_q\}, \quad (3.17)$$

$$\mathbf{0} \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}_b, \quad (3.18)$$

with

$$E_{\text{tot}}(\boldsymbol{\mu}) = \sum_{m=1}^{n_l} E_{1,m}(\boldsymbol{\mu}_{1,m}) + \sum_{m=1}^{n_c} E_{c,m}(\boldsymbol{\mu}_{c,m}), \quad (3.19)$$

where E_{tot} is the total energy consumption. $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_b$ are the optimization variable vector and its upper bound, respectively. Accordingly, n_l and n_c are the number of linear and corner

segments. g_r is the r^{th} equality constraint, q_y is the y^{th} inequality constraint. n_g and n_q are the number of equality and inequality constraints, respectively.

The extrema of the objective set in (3.15) can be derived by the independent minimization of each element as follows:

$$\begin{aligned} \varepsilon(\boldsymbol{\mu}_\varepsilon) &= 0, & E_{\max} &= E_{\text{tot}}(\boldsymbol{\mu}_\varepsilon), \\ \varepsilon_{\max} &= \varepsilon(\boldsymbol{\mu}_E), & E_{\min} &= E_{\text{tot}}(\boldsymbol{\mu}_E), \end{aligned} \quad (3.20)$$

with

$$\boldsymbol{\mu}_\varepsilon = \arg \min_{\boldsymbol{\mu}} \varepsilon(\boldsymbol{\mu}), \quad \boldsymbol{\mu}_E = \arg \min_{\boldsymbol{\mu}} E_{\text{tot}}(\boldsymbol{\mu})$$

The extrema in (3.20) are used to derive normalized objectives

$$\tilde{\varepsilon}(\boldsymbol{\mu}) = \frac{\varepsilon(\boldsymbol{\mu})}{\varepsilon_{\max}}, \quad \tilde{E}_{\text{tot}}(\boldsymbol{\mu}) = \frac{E_{\text{tot}}(\boldsymbol{\mu}) - E_{\min}}{\Delta E}, \quad (3.21)$$

with

$$\Delta E = E_{\max} - E_{\min},$$

being the energy saving potential. The Pareto optimal solutions of (3.15) can be transformed from a plane of cornering tolerance versus energy consumption with extrema coordinates $(0, E_{\max})$ and $(\varepsilon_{\max}, E_{\min})$ to a normalized plane with extrema points $\boldsymbol{\rho}_\varepsilon(0,1)$ and $\boldsymbol{\rho}_E(1,0)$ by using (3.21). Hence, (3.15) can be reformulated as:

$$\min_{\boldsymbol{\mu}} \tilde{E}_{\text{tot}}(\boldsymbol{\mu}), \quad (3.22)$$

subject to (3.16)–(3.18) and:

$$\mathbf{n}^T [\boldsymbol{\psi} - \boldsymbol{\rho}] \leq 0, \quad (3.23)$$

with

$$\begin{aligned}
\mathbf{n} &= \boldsymbol{\rho}_E - \boldsymbol{\rho}_\varepsilon = [1, -1]^T, \\
\boldsymbol{\psi} &= \left[\tilde{\varepsilon}(\boldsymbol{\mu}), \tilde{E}_{\text{tot}}(\boldsymbol{\mu}) \right]^T, \\
\boldsymbol{\rho} &= (1 - \zeta) \boldsymbol{\rho}_\varepsilon + \zeta \boldsymbol{\rho}_E, \\
0 &\leq \zeta \leq 1,
\end{aligned} \tag{3.24}$$

where ζ is a weighting factor. Every point $\boldsymbol{\rho}$ corresponds to an optimal solution $\boldsymbol{\psi}$.

A set of relevant Pareto optimal solutions Ψ can be obtained by using the divide and conquer algorithm in [36], where a minimum significance level between successive solutions is stipulated. The optimal solution with the best trade-off between the objectives

$$\boldsymbol{\psi}^* = \arg \min_{\boldsymbol{\psi}} \|\Psi\|_2, \tag{3.25}$$

corresponds to an assigned weighting factor ζ^* .

3.3.2 Trajectory constraints

A constraint set consisting of time equality, linear segment and smoothed corner segment constraints imposed on the optimization problem is described in this section. Without any loss of generality, a closed trajectory (i.e., a trajectory that has the same starting and ending point) with no corner smoothing at its start and end is considered. Hence, $n_c = n_l - 1$.

The time equality constraint

$$\sum_{m=1}^{n_l} T_{l,m}(\boldsymbol{\mu}_{l,m}) + \sum_{m=1}^{n_c} T_{c,m}(\boldsymbol{\mu}_{c,m}) - T_{\text{cycle}} = 0, \tag{3.26}$$

is applied to ensure that ψ satisfies the user-specified cycle time T_{cycle} . The m^{th} linear segment constraints are

$$\left. \begin{aligned} a_{\text{acc},k,m}^2 - a_{\text{lim},k}^2 &\leq 0 \\ a_{\text{dec},k,m}^2 - a_{\text{lim},k}^2 &\leq 0 \\ v_{\text{con},k,m}^2 - v_{\text{lim},k}^2 &\leq 0 \end{aligned} \right\}, \forall m, \quad (3.27)$$

$$\left. \begin{aligned} a_{\text{s},l,k,m} &= 0 \\ a_{\text{e},l,k,m} - a_{\text{s},c,k,m} &= 0 \\ v_{\text{s},l,k,m} &= 0 \\ v_{\text{e},l,k,m} - v_{\text{s},c,k,m} &= 0 \\ \Delta s_{l,k,m} + [L_{c,m} - L_m] h_{\text{s},k,m} &= 0 \end{aligned} \right\}, m = 1, \quad (3.28)$$

$$\left. \begin{aligned} a_{\text{s},l,k,m} - a_{\text{e},c,k,m-1} &= 0 \\ a_{\text{e},l,k,m} - a_{\text{s},c,k,m} &= 0 \\ v_{\text{s},l,k,m} - v_{\text{e},c,k,m-1} &= 0 \\ v_{\text{e},l,k,m} - v_{\text{s},c,k,m} &= 0 \\ \Delta s_{l,k,m} + [L_{c,m-1} + L_{c,m} - L_m] h_{\text{s},k,m} &= 0 \end{aligned} \right\}, m \neq [1, n_1], \quad (3.29)$$

$$\left. \begin{aligned} a_{\text{s},l,k,m} - a_{\text{e},c,k,m-1} &= 0 \\ a_{\text{e},l,k,m} &= 0 \\ v_{\text{s},l,k,m} - v_{\text{e},c,k,m-1} &= 0 \\ v_{\text{e},l,k,m} &= 0 \\ \Delta s_{l,k,m} + [L_{c,m-1} - L_m] h_{\text{s},k,m} &= 0 \end{aligned} \right\}, m = n_1, \quad (3.30)$$

$$j_{\text{max},k,m} - j_{\text{lim},k,m} = 0, \forall m, \quad (3.31)$$

Table 3.1: Identified energy consumption model parameters

| k^{th} axis | $c_{1,k}$ Ws^4/m^2 | $c_{2,k}$ Ws^2/m^2 | $c_{3,k}$ Ws/m | $c_{4,k}$ W | $c_{5,k}$ Ws^2/m | $c_{6,k}$ Ws^3/m^2 |
|---------------|----------------------|----------------------|------------------|---------------|--------------------|----------------------|
| 1 | 2.264 | 793.589 | 82.567 | 1.636 | 3.849 | 97.122 |
| 2 | 1.859 | 1124.057 | 78.894 | 1.142 | 2.914 | 95.564 |

where $j_{lim,k,m}$, $a_{lim,k}$, and $v_{lim,k}$ are the limits for jerk, acceleration and velocity, respectively. The smoothed corner segment nodes are indexed by s for the start and e for its end. Linear and corner segments are indexed by l, and c, respectively. The m^{th} linear segment length before corner smoothing is given by L_m . The m^{th} smoothed corner segment constraints are

$$\left. \begin{aligned} j_{c,k,m}^2 - j_{lim,k,m}^2 &\leq 0 \\ (j_{c,k} T_{1,c})^2 - a_{lim,k,m}^2 &\leq 0 \\ v_{s,c,k,m}^2 - v_{lim,k,m}^2 &\leq 0 \\ v_{e,c,k,m}^2 - v_{lim,k,m}^2 &\leq 0 \\ \|\mathbf{p}_{c,m} - \mathbf{p}_{mid,m}\|_2 - \varepsilon &= 0 \end{aligned} \right\}, \forall m. \quad (3.32)$$

SQP [9, 33, 62] is utilized to solve the optimization problem.

3.4 Optimization results and discussions

The parameters of the energy consumption model are obtained by the identification of an actual industrial biaxial feed drive system (Section 2.1.3). Table 3.1 shows the identified model coefficients.

The contour in Fig. 3.2 is used to generate Pareto optimal corner smoothed trajectories at a cycle time $T_{cycle} = 5.227s$ and kinematic limits: $j_{lim,k} = 200,000 \text{ mm/s}^3$, $a_{lim,k} = 500 \text{ mm/s}^2$ and $v_{lim,k} = 80 \text{ mm/s}$. Since the minimum energy is achieved at a maximum cornering tolerance $\varepsilon_{max} = 1248.6 \mu\text{m}$, the upper bound cornering tolerance ε_{ub} is set to $100 \mu\text{m}$ to limit the solution set to within practical margins. Solutions for the optimization problem in Section 3.3 are obtained by using SQP in a MATLAB environment on a laptop computer with 2.50 GHz CPU,

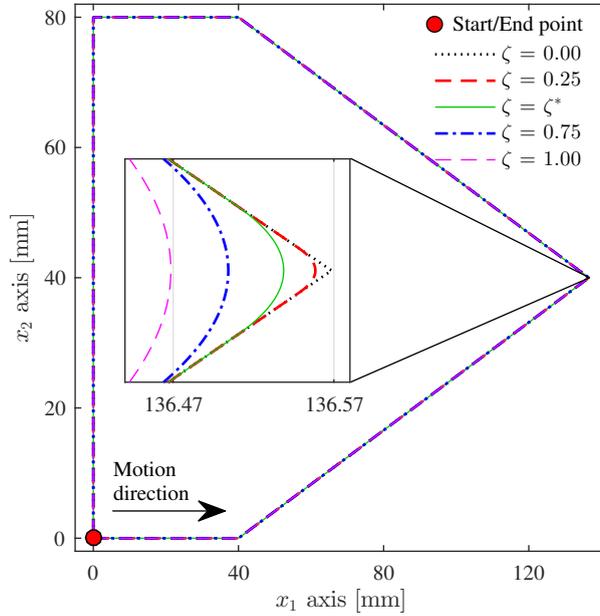


Fig. 3.2: Contour used for optimal trajectory generation.

8 GB RAM and Windows 10 operating system. A significance level of 0.02 is used in the divide and conquer algorithm to generate a Pareto front of the optimization results.

The zoomed portion in Fig. 3.2 shows smoothed corners generated as part of the Pareto optimal solutions at different weighting factors ζ . $\zeta = 0.00$ and $\zeta = 1.00$ correspond to the cornering tolerances of $\varepsilon = 0 \mu\text{m}$ and $\varepsilon = 100 \mu\text{m}$, respectively. The best trade solution is obtained at $\zeta^* = 0.469$.

The Pareto front of the optimal solutions is shown in Fig. 3.3. A trade-off between the energy consumption and cornering tolerance is observed, where the trajectory consisting of point-to-point (PTP) movements (i.e., $\varepsilon = 0 \mu\text{m}$) consumes the maximum amount of energy. The energy-saving potential is 1.3 J. The best trade-off solution occurs at a cornering tolerance of $\sim 30 \mu\text{m}$, where it achieves $\sim 64\%$ of energy-saving potential.

Fig. 3.4 shows the generated path velocity profiles at different weighting factors ζ . At each corner, it is observed that cornering velocity increases with the increase in ζ . This is because lax cornering tolerances allow fast cornering velocities and vice-versa.

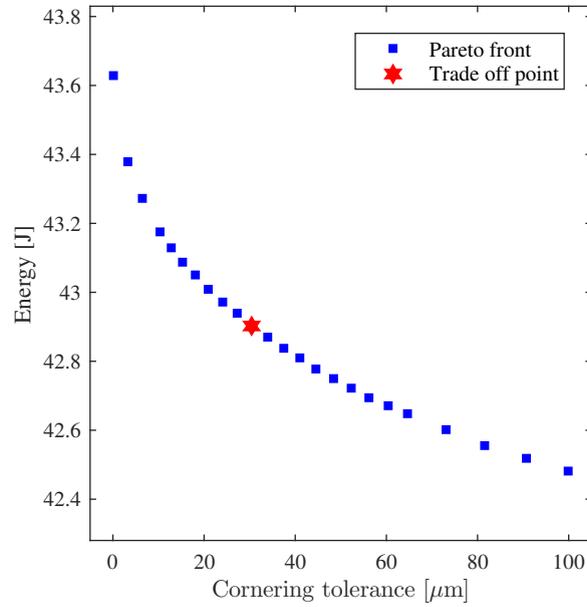


Fig. 3.3: Pareto front representation of optimization results

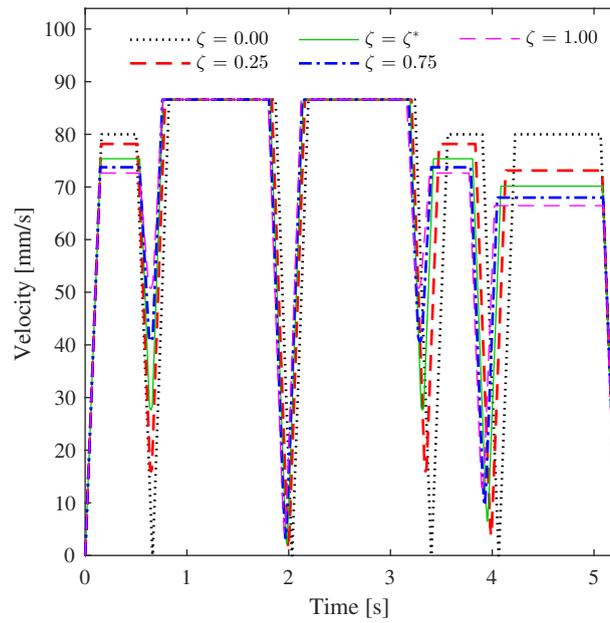


Fig. 3.4: Generated velocity profiles at different optimal weighting factors.

More energy can be saved by increasing ε_{ub} although this is detrimental to the quality of the corners. Fig. 3.5 shows the trend of variation of the best trade-off cornering tolerances and energy-saving potential with ε_{ub} .

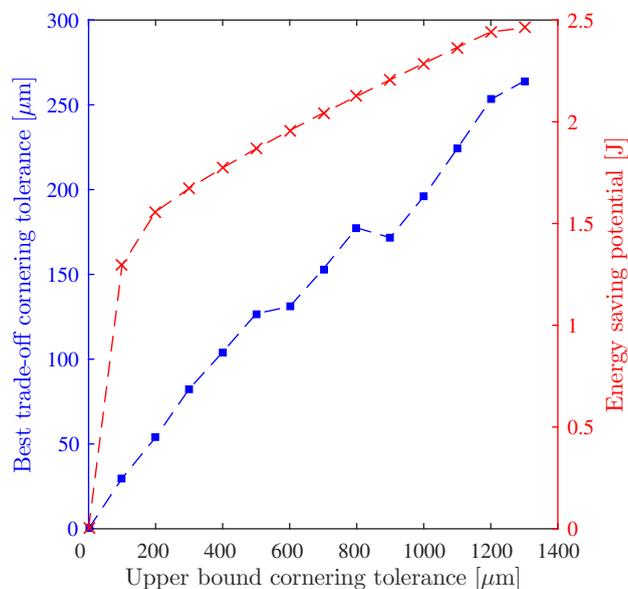


Fig. 3.5: Variation of best trade-off cornering tolerance and energy saving potential under upper bound of cornering tolerance.

3.5 Summary

This chapter presents a method of generating piece-wise linear contours with smoothed corners by a trade-off between energy consumption and cornering error. Linear segments and cornering paths are defined using JLAP and KCSIA, respectively. A BOOP is solved and the best trade-off cornering error is acquired under a specified cycle time as a constraint. The best trade-off point on the Pareto frontier provides the best compromise between energy consumption and cornering error.

Chapter 4

A Trade-off between Energy Saving and Cycle Time Reduction by Pareto-optimal Corner Smoothing

4.1 Introduction

CNC machine tools are extensively used in the manufacturing industry due to their accuracy, precision and speed in performing repetitive tasks [95]. The objectives of improving accuracy and precision, productivity and energy saving motivate many research types on feed drive systems.

Several studies have been conducted to achieve precise motion, where the quality of a reference trajectory influences the resulting motion's performance. Barre et al. show the influence of jerk on the vibrational behavior of industrial machines [6]. Sencer and Tajima propose a generation of frequency optimal acceleration profiles for suppressing vibrations [75].

The productivity of typical repetitive tasks in the manufacturing industry, such as standard component machining or a combination of parts in an assembly line, can be increased by cycle time reduction. The productivity of repetitive motions can be improved by reducing cycle time. An analytical solution for minimizing motion durations with JLAPs is proposed by

Joeng et al. [44]. Altintas and Erkorkmaz propose a time-optimal feed rate scheduling method for quintic spline tool path interpolation [2]. Mori et al. propose a time-optimal trajectory planning method for traversing contour curves by considering second-order dynamics of feed drive systems [57]. Dong et al. propose a time optimizing algorithm for scheduling jerk-limited feed-rates for a given parametric curve [25]. Erkorkmaz and Heng propose the optimization of S-curve feed profiles for non-uniform rational B-spline tool paths by a heuristic search approach to minimize cycle time [29]. For robotic manipulators, Uchiyama et al. propose a time-optimal trajectory generation approach that considers actuator power limits and obstacles in a work environment [93].

Concerns on carbon footprint and the need for reducing production costs have catalyzed the research on energy saving. Although controller designs that consider energy saving have been proposed in literature [26, 61], optimal trajectory planning is an advantageous alternative, especially for industrial systems with inaccessible controllers. Park proposes an approach for generating energy-efficient velocity profiles for electromechanical systems with repetitive PTP motions [66]. Halevi et al. propose energy minimal trajectory generation in redundantly actuated machine tools for a predefined end-effector path [35]. Analytical solutions for energy saving using S-curve PTP trajectories with cycloid motions during acceleration periods for a given motion time and distance have been proposed in [40]. This approach is extended for S-curve PTP trajectories with trapezoidal acceleration periods in [91]. Oda et al. propose a method of optimization rotation angles and feed rate profiles along curvilinear paths to minimize energy consumption [65]. Bi et al. propose an approach for determining the energy-optimal tool holder poses for parallel kinematic machine tools [8].

The objectives of minimizing cycle time and energy consumption are contradictory. Diaz et al. describe a parabolic relation between energy consumption and feed-rate since the power demand consists of constant and feed-rate dependent terms [24]. Thus, these objectives require MOOP. Yan and Li propose the optimization of cutting parameters in a milling process to trade-off among energy consumption, material removal rate and surface roughness based on a weighted grey relational analysis [104]. He et al. propose a method of determining optimal machining parameters from a cost function consisting of a weighted sum of normalized energy consumption, cutting force and processing time objectives using Pareto frontiers generated by evolutionary algorithms [37].

Although minimizing a weighted sum of objectives is the standard method of generating Pareto frontiers in MOOPs, it has a drawback of producing a non-uniform distribution of Pareto points even from a uniform set of weights, as illustrated by Das and Dennis in [20]. In order to counter the drawbacks of weighted sum, Das and Dennis propose NBI for solving MOOPs by reducing them into a set of SOOPs [21]. The NBI method involves forming a CHIM on a normalized objective space, constructing evenly distributed quasi-normal lines to the CHIM and maximizing the CHIM distance towards the utopia point along each quasi-normal line. Every solution corresponds to a Pareto optimal point. The NBI method has a drawback of producing non-Pareto optimal points in some cases.

Messac et al. propose the NNC method for solving a MOOP [55]. Like the NBI method, the NNC method constructs a CHIM with evenly distributed points in normalized objective space. At every CHIM point, the MOOP is reduced to a SOOP, where a normalized objective is minimized while the remaining objectives are incorporated as additional inequality constraints. The NNC method is less prone to generating non-Pareto optimal points compared to the NBI method. The NNC method produces evenly distributed Pareto optimal points for a bi-objective case, but this is not guaranteed for more than two objectives.

Sanchis et al. propose the enhanced NNC method that produces evenly distributed Pareto optimal points for MOOPs [69]. Although the problem formulation is similar to the NNC method, the enhanced NNC method normalizes the objectives by either an exact linear transformation or mapping using pseudo-anchor points, where a pseudo-anchor point is a point in an objective space that minimizes one objective but maximizes the rest.

Evenly distributing points across a Pareto front has a drawback of including points that are not significant to a decision-maker. Hence, Pareto point filtering would be required according to a prespecified trade-off level. To avoid the computational cost of generating insignificant solutions, Hashem et al. propose the divide and conquer algorithm for obtaining a Pareto front with adaptive resolution [36]. Using the NBI or NNC problem formulation, the divide and conquer algorithm recursively explores the CHIM for significant solutions. Insignificant regions of the Pareto front are left unexplored according to a prespecified trade-off level.

Local corner smoothing algorithms have been proposed to improve machining quality ([41, 106]) and reduce cycle time for motion along piecewise linear contours. Sencer et al. propose blending discontinuous axis velocity commands using curvature optimal quintic Bézier curves to

reduce cycle time [72]. Local corner smoothing algorithms that use parametric curves such as B-splines [41, 106] and Bézier curves [72] solve a local corner smoothing problem in two steps: curve fitting proceeded by feed profile planning. This approach is inefficient since parametric curves suffer computational drawbacks in real-time interpolation [28, 72, 89]. Since KCS methods do not require parametric curve fitting, they are proposed in the literature to overcome these drawbacks. Sencer et al. propose a KCS method that smoothly blends axis velocities using FIR filters to reduce cycle time and suppress vibration [73]. Tajima and Sencer propose KCSIA and KCSUA for analytically computing C^2 continuous feed motions while exploiting axis limits to achieve time optimality [84, 85]. In [84], KCSIA and KCSUA produce better contouring performances and shorter cycle times than Bézier curves [72]. The Bézier curve's inferior contouring performance is attributed to the lack of jerk limitation at corners, resulting in the excitation of a feed drive system's vibratory dynamics. Despite the good performance in reducing cycle time, the potential of KCS in energy saving is not explored in [84] nor [85]. For a known energy consumption model, Nshama et al. illustrated a bi-objective optimization of energy consumption and cycle time for corners smoothed by KCSIA [64]. However, the authors' methods of model identification and experimental verification of the optimization results on an industrial feed drive system are not provided.

This work proposes the generation and experimental verification of Pareto optimal trajectories that trade-off cycle time with energy consumption in light of the preceding. Piecewise linear contours are considered in this study since such paths are typically used in the manufacturing industry. Before formulating and solving the bi-objective optimization problem, an energy consumption model of an industrial feed drive system is identified by the LSE method. In this work, energy consumption is considered an electrical utility, where the system is represented as an electrical load. Thus, energy consumption is defined as the electrical energy drawn by the system during trajectory tracking. This energy is assumed to be partly stored as kinetic energy while the rest of it is dissipated as electrical and mechanical losses. In this work, it is assumed that the system is non-regenerative. In the optimization problem, a trajectory along a linear segment is described as a JLAP [27] while motion transition at a corner is smoothed using two cases: KCSIA and KCSUA. The optimization problem is formulated using the NNC method, where a solution consists of optimal parameters for each line and smoothed corner. Pareto frontiers for KCSIA and KCSUA are generated using the divide and conquer algorithm, where each solution is computed using SQP [9, 33, 62]. The best trade-off solution on a Pareto

frontier is selected as the one that is nearest to the utopia point. Optimization results show that KCSIA offers the best trade-off. These results are validated experimentally.

In summary, the contributions of this work are as follows:

- An energy consumption model of an industrial feed drive system is identified by LSE and validated experimentally.
- Pareto-optimal local corner smoothing with KCSIA and KCSUA is proposed.
- Optimization and repetitive experimental results show that KCSIA produces the best trade-off solution.

The next section describes the identification method of an energy consumption model. Trajectory descriptions are illustrated in Section 4.3, followed by the optimization problem formulation in Section 4.4. Pareto optimal solutions and their experimental verification are presented in Section 4.5, followed by concluding remarks in Section 4.6.

4.2 Identification of the energy consumption model

A method of identifying the energy model parameters is described in this section. After that, a setup for the identification experiment and its results are presented.

4.2.1 Model identification

The energy coefficients in Section 2.2 are required to generate optimal trajectories. Since the energy consumption model (2.12) at different loading conditions can be derived from the no-load energy consumption model (2.9) provided the load mass is known as shown in (2.10) and (2.13), model identification at no-load conditions is considered in this study without any loss in generality. It is assumed that the power consumption can be measured, where these measurements can only be affected by white noise with zero mean. LSE is used for energy

model identification, whereby the estimated power and estimation error at the j^{th} data update instant for each axis are respectively given by

$$\begin{aligned}\hat{P}_{k,j} &= c_{1,k}\dot{x}_{k,j}^2 + c_{2,k}\ddot{x}_{k,j}^2 + c_{3,k}\dot{x}_{k,j}\text{sgn}(\dot{x}_{k,j}) + c_{4,k} + c_{5,k}\ddot{x}_{k,j}\text{sgn}(\dot{x}_{k,j}) + c_{6,k}\ddot{x}_{k,j}\dot{x}_{k,j}, \\ e_{k,j} &= \hat{P}_{k,j} - P_{k,j},\end{aligned}\quad (4.1)$$

where $P_{k,j}$ is the measured power for the k^{th} axis at data point j . The LSE minimization problem is defined as

$$\min_{\mathbf{c}_k} \sum_{j=1}^{N_p-1} (e_{k,j+1}^2 + e_{k,j}^2), \quad (4.2)$$

subject to

$$\begin{aligned}c_{6,k}^2 - 4c_{1,k}c_{2,k} &\geq 0, \\ c_{5,k}^2 - 4c_{1,k}c_{4,k} &= 0,\end{aligned}\quad (4.3)$$

$$\begin{aligned}c_{4,k}c_{6,k} - c_{1,k}c_{2,k} &= 0, \\ \mathbf{c}_k &> \mathbf{0},\end{aligned}\quad (4.4)$$

with

$$\mathbf{c}_k = [c_{1,k}, c_{2,k}, \dots, c_{6,k}]^{\text{T}}, \quad (4.5)$$

where N_p is the number of data points. The objective function in (4.2) represents a numerical integration of estimation error squares described by trapezoidal rule. (4.3) is obtained from rearranging energy coefficients in (2.10). (4.4) represents the bound constraints of the coefficients. SQP, in a MATLAB[®] environment, is used to solve (4.2) on a Windows 10 laptop computer with the specifications: core i7 intel processor, 2.50 GHz CPU and 8 GB RAM.

4.2.2 Identification results

The identification experiment is conducted on both axes of the industrial biaxial table (Section 2.1.3) whereby a reference trajectory consisting of several constant acceleration (i.e., 20, 40,

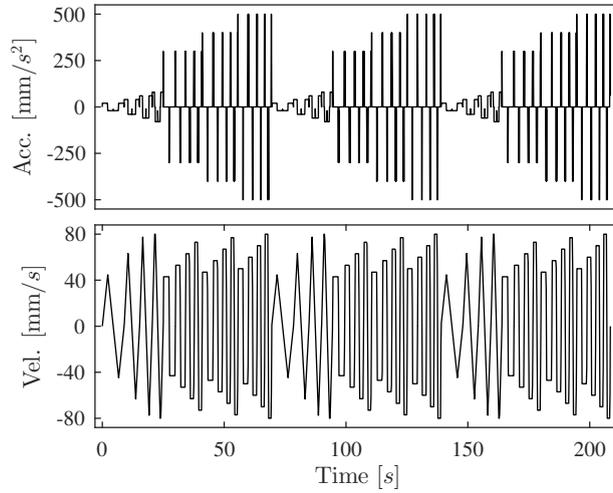


Fig. 4.1: Reference acceleration and velocity profiles used for the identification experiment.

Table 4.1: Identified energy coefficients

| k^{th} axis | \dot{x}_k | $c_{1,k}$ Ws^4/m^2 | $c_{2,k}$ Ws^2/m^2 | $c_{3,k}$ Ws/m | $c_{4,k}$ W | $c_{5,k}$ Ws^2/m | $c_{6,k}$ Ws^3/m^2 |
|----------------------|--------------------|------------------------------------|------------------------------------|--------------------------------|----------------------|----------------------------------|------------------------------------|
| 1 | $\dot{x}_1 \geq 0$ | 2.684 | 546.357 | 45.135 | 0.663 | 2.667 | 90.838 |
| | $\dot{x}_1 < 0$ | 2.684 | 476.807 | -52.333 | 0.891 | -3.093 | 90.838 |
| 2 | $\dot{x}_2 \geq 0$ | 2.101 | 534.437 | 50.851 | 0.856 | 2.682 | 79.682 |
| | $\dot{x}_2 < 0$ | 2.101 | 494.006 | -48.999 | 0.795 | -2.584 | 79.682 |

60, 80, 300, 400 and 500 mm/s^2) and velocity (i.e., 40, 43, 47, 50, 53, 57, 60, 63, 67, 70, 73, 77 and 80 mm/s) profiles, shown in Fig. 4.1, is used. The experiment is conducted for 15 iterations, whereby power measurements and velocity data are used to solve (4.2). Since a numerically computed acceleration is noisy, the reference acceleration profile is used instead in solving (4.2). Fig. 4.2 shows the matching between measured and estimated powers in each axis. Fig. 4.3 shows a comparison between energy values computed from power measurements and those estimated by the model using the velocity data. The corresponding identified energy coefficients are given in Table 4.1.

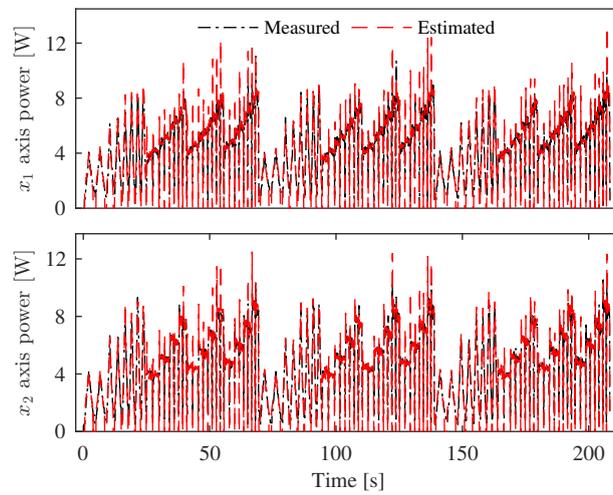


Fig. 4.2: Measured and estimated power consumption during the identification experiment. The measured power is labeled as ‘Measured’ while the power estimated by the model is labeled as ‘Estimated’.

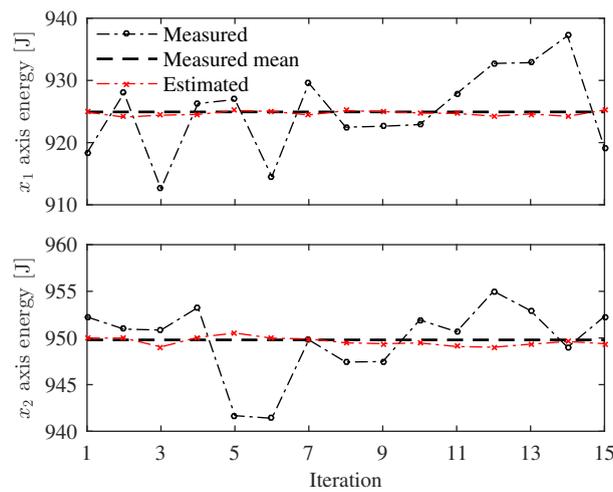


Fig. 4.3: Measured and estimated energy consumption during identification experiment. The energy computed from power measurements and model estimates are labeled as ‘Measured’ and ‘Estimated’, respectively.

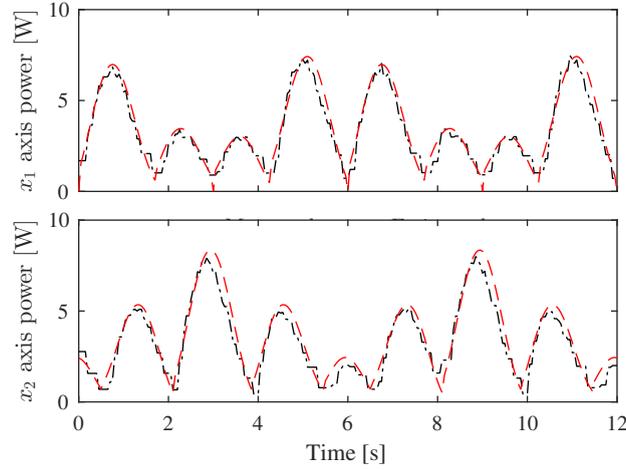


Fig. 4.4: Measured and estimated power consumption during the verification experiment. The measured power is labeled as ‘Measured’ while the power computed by the model is labeled as ‘Estimated’.

4.2.3 Model verification

A trifolium trajectory

$$\begin{aligned}
 \mathbf{x}(t) &= r^*(t) \left[\cos\left(\frac{2\pi t}{T}\right), \sin\left(\frac{2\pi t}{T}\right) \right]^\top, \\
 r^*(t) &= r \cos\left(\frac{2\pi t}{T}\right) \left\{ 4\sin^2\left(\frac{2\pi t}{T}\right) - 1 \right\}, \\
 0 &\leq t \leq T,
 \end{aligned} \tag{4.6}$$

is used to test validity of the model, where r and T are the radius (=50 mm) and period (=12sec) of the motion. A verification experiment is conducted for 10 iterations. Fig. 4.4 shows the measured and estimated power consumption for the 9th iteration. The estimation is done by inserting the trajectory velocity and acceleration, derived from (4.6), into (2.9). The consumed energy measurement results are shown in Fig.4.5, where the average percentage energy errors relative to the model estimates are $\sim -7\%$ and $\sim -10\%$ for x_1 and x_2 axes, respectively. Since the model estimates of power and energy consumption are similar to the experimental ones, the model can be used for generating energy optimal trajectories.

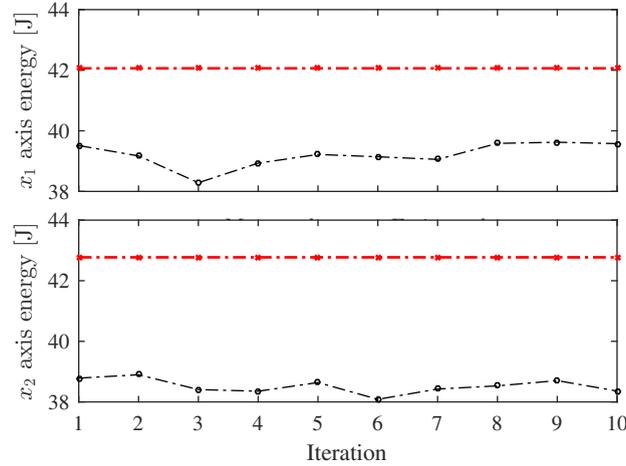


Fig. 4.5: Measured and estimated energy consumption during verification experiment. The energy computed from power measurements and model estimates are labeled as ‘Measured’ and ‘Estimated’, respectively.

4.3 Trajectory representation

This section describes trajectory generation methods for geometries consisting of piecewise linear segments and smoothed corners. Linear segment trajectories are defined by JLAP, while smoothed cornering paths are described using KCS.

4.3.1 Jerk-limited acceleration profile

JLAP is a trajectory profile that imposes a smooth velocity transition between points by restricting the jerk, acceleration and velocity within predefined limits [27, 84]. The trajectory consists of an acceleration phase from t_0 to $t_{3,a}$, a constant velocity phase from $t_{3,a}$ to t_{con} and

a deceleration phase from t_{con} to t_1 . The trajectory jerk is given as follows:

$$\ddot{x}_{1,k}(t) = \begin{cases} j_{\max,k}, & t_{0,l} \leq t < t_{1,a}, \\ 0, & t_{1,a} \leq t < t_{2,a}, \\ -j_{\max,k}, & t_{2,a} \leq t < t_{3,a}, \\ 0, & t_{3,a} \leq t < t_{\text{con}} \\ -j_{\max,k}, & t_{\text{con}} \leq t < t_{1,d}, \\ 0, & t_{1,d} \leq t < t_{2,d}, \\ j_{\max,k}, & t_{2,d} \leq t < t_1, \end{cases} \quad (4.7)$$

where $j_{\max,k}$ is the k^{th} axial jerk limit. The time intervals

$$\begin{aligned} T_{1,a} &= t_{1,a} - t_{0,l}, & T_{1,d} &= t_{1,d} - t_{\text{con}}, \\ T_{2,a} &= t_{2,a} - t_{1,a}, & T_{2,d} &= t_{2,d} - t_{1,d}, \\ T_{3,a} &= t_{3,a} - t_{2,a}, & T_{3,d} &= t_1 - t_{2,d}, \\ T_{\text{con}} &= t_{\text{con}} - t_{3,a}, \end{aligned} \quad (4.8)$$

are determined such that the trajectory satisfies kinematic limits (i.e., jerk, acceleration and velocity limits) and the geometric constraints. The time intervals in (4.8) define a parameter vector

$$\boldsymbol{\mu}_1 = [T_{1,a}, T_{2,a}, T_{3,a}, T_{\text{con}}, T_{1,d}, T_{2,d}, T_{3,d}], \quad (4.9)$$

which can be used to optimize the trajectory, where the total duration is given by

$$T_1(\boldsymbol{\mu}_1) = T_{1,a} + T_{2,a} + T_{3,a} + T_{\text{con}} + T_{1,d} + T_{2,d} + T_{3,d}, \quad (4.10)$$

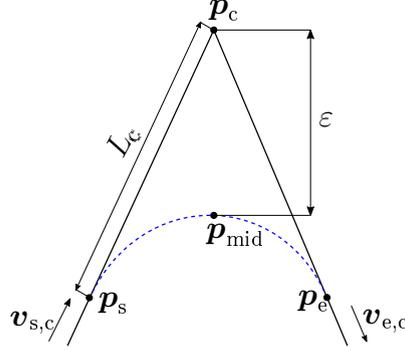


Fig. 4.6: Geometric description of a cornering path using the KCS method.

and the energy consumed in traversing the trajectory is obtained by inserting acceleration and velocity expressions, which are acquired from successive integrations of (4.7), into (2.9) to get

$$\begin{aligned}
 E_1(\mu_1) = & \int_{t_{0,1}}^{t_1(\mu_1)} \ddot{\mathbf{x}}_1(t)^\top \mathbf{C}_1 \ddot{\mathbf{x}}_1(t) + \dot{\mathbf{x}}_1(t)^\top \mathbf{C}_2 \dot{\mathbf{x}}_1(t) \\
 & + \dot{\mathbf{x}}_1(t)^\top \mathbf{C}_3 \text{sgn}(\dot{\mathbf{x}}_1(t)) + \text{tr}(\mathbf{C}_4) \\
 & + \ddot{\mathbf{x}}_1(t)^\top \mathbf{C}_5 \text{sgn}(\dot{\mathbf{x}}_1(t)) + \ddot{\mathbf{x}}_1(t)^\top \mathbf{C}_6 \dot{\mathbf{x}}_1(t) dt.
 \end{aligned} \tag{4.11}$$

4.3.2 Kinematic corner smoothing

KCS is a 2D method that blends axial velocities at corners of piecewise linear tool paths by analytically computing the cornering velocity, acceleration and duration, based on JLAP, while taking into account a user-specified tolerance $0 < \varepsilon \leq \varepsilon_{ub}$ and axial limits [84]. This produces a cornering trajectory that has a smooth motion transition from the end of one linear segment to the start of the next. Fig. 4.6 shows an illustration of a cornering path from position vector \mathbf{p}_s to \mathbf{p}_e where the maximum cornering error ε is located at the mid-point \mathbf{p}_{mid} . \mathbf{p}_c is the position of the original corner. The magnitude of tangential velocities at the start and end of the trajectory are set to be the same (i.e., $\|\mathbf{v}_{s,c}\|_2 = \|\mathbf{v}_{e,c}\|_2 = V_c$). Two KCS methods have been proposed in [84]: KCSIA and KCSUA described below.

4.3.2.1 Kinematic corner smoothing with interrupted acceleration

KCSIA generates cornering trajectories while setting the tangential acceleration at the start and end of the motion to zero. The trajectory is described as follows:

$$\ddot{x}_{c,k}(t) = \begin{cases} j_{c,k}, & t_{0,c} \leq t < t_{1,c} \\ 0, & t_{1,c} \leq t < t_{2,c} \\ -j_{c,k}, & t_{2,c} \leq t < t_c, \end{cases} \quad (4.12)$$

with

$$\begin{aligned} j_{c,k} &= \frac{v_{e,c,k} - v_{s,c,k}}{T_{1,c}(T_{1,c} + T_{2,c})}, \\ v_{s,c,k} &= V_c h_{s,k}, & v_{e,c,k} &= V_c h_{e,k}, \\ T_{1,c} &= t_{1,c} - t_{0,c} = t_c - t_{2,c}, & T_{2,c} &= t_{2,c} - t_{1,c}, \\ a_{1,c,k} &= j_{c,k} T_{1,c}, \end{aligned} \quad (4.13)$$

where $j_{c,k}$ is the cornering jerk. $h_{s,k}$ and $h_{e,k}$ are the direction vector elements along the k^{th} axis at the trajectory starting and terminal points, respectively. The total displacement in each axis $\Delta x_{c,k}$ can be obtained by integration of (4.12):

$$\Delta x_{c,k} = \left[v_{s,c,k} + \frac{1}{2} j_{c,k} T_{1,c} (T_{1,c} + T_{2,c}) \right] (2T_{1,c} + T_{2,c}). \quad (4.14)$$

By considering one axis, the Euclidean length L_c used for corner smoothing can be expressed as

$$L_c = \frac{\Delta x_{c,k}}{(h_{s,k} + h_{e,k})}. \quad (4.15)$$

The components of position vectors \mathbf{p}_s , \mathbf{p}_{mid} and \mathbf{p}_e can be obtained by using (4.15) as follows:

$$\begin{aligned} p_{s,k} &= p_{c,k} - L_c h_{s,k}, \\ p_{\text{mid},k} &= p_{s,k} + v_{s,c,k} \left[T_{1,c} + \frac{1}{2} T_{2,c} \right] + j_{c,k} \left[\frac{1}{2} T_{1,c} \left(\frac{T_{2,c}}{2} \right)^2 \right] + j_{c,k} \left[\frac{1}{6} T_{1,c}^3 + \frac{1}{2} T_{1,c}^2 \left(\frac{T_{2,c}}{2} \right) \right], \\ p_{e,k} &= p_{c,k} + L_c h_{e,k}, \end{aligned} \tag{4.16}$$

whereby, a cornering constraint

$$\|\mathbf{p}_c - \mathbf{p}_{\text{mid}}\|_2 - \varepsilon_b \leq 0, \tag{4.17}$$

is defined. \mathbb{C}^2 continuity of motion transition from the end of the m^{th} linear segment to the start of the $(m+1)^{\text{th}}$ linear segment is established by setting

$$\begin{aligned} a_{1,m,k} = \ddot{x}_{c,k}(t_{0,c}) = 0, & \quad \ddot{x}_{c,k}(t_c) = a_{0,m+1,k} = 0, \\ v_{1,m,k} = v_{s,c,k}, & \quad v_{e,c,k} = v_{0,m+1,k}, \\ p_{1,m,k} = p_{s,k}, & \quad p_{e,k} = p_{0,m+1,k}, \end{aligned} \tag{4.18}$$

where $a_{1,m,k}$, $v_{1,m,k}$ and $p_{1,m,k}$ are the k^{th} axis acceleration, velocity and position at the end of the m^{th} linear segment. $a_{0,m+1,k}$, $v_{0,m+1,k}$ and $p_{0,m+1,k}$ are the k^{th} axis acceleration, velocity and position at the start of the $(m+1)^{\text{th}}$ linear segment.

The cornering trajectory can be described by a parameter vector

$$\boldsymbol{\mu}_c = [V_c, T_{1,c}, T_{2,c}], \tag{4.19}$$

provided that the geometry, axial limits and cornering tolerance are specified, as shown in (4.12)-(4.18). In a similar manner to Section 4.3.1, the trajectory duration and energy consumption

are respectively given by

$$T_c = 2T_{1,c} + T_{2,c}, \quad (4.20)$$

$$E_c(\mu_c) = \int_{t_{0,c}}^{t_c(\mu_c)} \ddot{\mathbf{x}}_c(t)^\top \mathbf{C}_1 \ddot{\mathbf{x}}_c(t) + \dot{\mathbf{x}}_c(t)^\top \mathbf{C}_2 \dot{\mathbf{x}}_c(t) + \dot{\mathbf{x}}_c(t)^\top \mathbf{C}_3 \text{sgn}(\dot{\mathbf{x}}_c(t)) + \text{tr}(\mathbf{C}_4) \\ + \ddot{\mathbf{x}}_c(t)^\top \mathbf{C}_5 \text{sgn}(\dot{\mathbf{x}}_c(t)) + \ddot{\mathbf{x}}_c(t)^\top \mathbf{C}_6 \dot{\mathbf{x}}_c(t) dt. \quad (4.21)$$

4.3.2.2 Kinematic corner smoothing with uninterrupted acceleration

KCSUA produces jerk limited cornering trajectories by introducing non-zero tangential accelerations of equal magnitude A_c at the starting and terminal points of the motion. The trajectory is expressed as

$$\ddot{x}_{c,k}(t) = j_{c,k}, \quad t_{0,c} \leq t < t_{1,c}, \quad (4.22)$$

with

$$j_{c,k} = \frac{a_{e,c,k} - a_{s,c,k}}{T_{1,c}}, \quad T_{1,c} = t_{1,c} - t_{0,c}, \\ a_{s,c,k} = -A_c h_{s,k}, \quad a_{e,c,k} = A_c h_{e,k}, \\ v_{s,c,k} = V_c h_{s,k}, \quad v_{e,c,k} = V_c h_{e,k}, \quad (4.23)$$

where $a_{s,c,k}$ and $a_{e,c,k}$ are the accelerations at the start and end of the corner trajectory, respectively.

From (4.22), the relationship between V_c and A_c is obtained as

$$V_c = \frac{A_c T_{1,c}}{2}. \quad (4.24)$$

The total displacement in the k^{th} axis

$$\Delta x_{c,k} = v_{s,c,k} T_{1,c} + \frac{1}{2} a_{s,c,k} T_{1,c}^2 + \frac{1}{6} j_{c,k} T_{1,c}^3, \quad (4.25)$$

is obtained by integrating (4.22). From (4.15) and (4.25), the position vector components

$$\begin{aligned} p_{s,k} &= p_{c,k} - L_c h_{s,k}, \\ p_{\text{mid},k} &= p_{s,k} + v_{s,c,k} \left(\frac{T_{1,c}}{2} \right) + \frac{1}{2} a_{s,c,k} \left(\frac{T_{1,c}}{2} \right)^2 + \frac{1}{6} j_{c,k} \left(\frac{T_{1,c}}{2} \right)^3, \\ p_{e,k} &= p_{c,k} + L_c h_{e,k}, \end{aligned} \quad (4.26)$$

can be derived. The cornering constraint for KCSUA is also expressed by (4.17). \mathbb{C}^2 continuity of motion transition at the start and end of the cornering trajectory is established by the following conditions with regard to the m^{th} and $(m+1)^{\text{th}}$ linear segments

$$\begin{aligned} a_{1,m,k} &= a_{s,c,k}, & a_{e,c,k} &= a_{0,m+1,k}, \\ v_{1,m,k} &= v_{s,c,k}, & v_{e,c,k} &= v_{0,m+1,k}, \\ p_{1,m,k} &= p_{s,k}, & p_{e,k} &= p_{0,m+1,k}. \end{aligned} \quad (4.27)$$

(4.22)-(4.27) shows that a KCSUA trajectory can be defined by a parameter vector

$$\boldsymbol{\mu}_c = [A_c, T_{1,c}], \quad (4.28)$$

for given a geometry, kinematic limits and cornering tolerance. The duration of trajectory is obtained as:

$$T_c = T_{1,c}. \quad (4.29)$$

The energy consumed in the cornering motion is also expressed using (4.21).

4.4 Trajectory optimization

This section illustrates a method of determining optimal trajectories that trade-off between energy consumption and cycle time for paths consisting of piecewise linear segments and smoothed corners. A bi-objective optimization problem and its constraints are described below.

4.4.1 Problem formulation

The mathematical representation of the optimization problem is described as follows:

$$\begin{aligned} \min_{\boldsymbol{\mu}} \{T_{\text{tot}}(\boldsymbol{\mu}), E_{\text{tot}}(\boldsymbol{\mu})\}, \\ \boldsymbol{\mu} = [\boldsymbol{\mu}_{1,1}, \boldsymbol{\mu}_{c,1}, \boldsymbol{\mu}_{1,2}, \boldsymbol{\mu}_{c,2}, \dots, \boldsymbol{\mu}_{1,n_1}, \boldsymbol{\mu}_{c,n_c}]^{\top}, \end{aligned} \quad (4.30)$$

subject to

$$\begin{aligned} g_r(\boldsymbol{\mu}) &= 0, & r &\in \{1, 2, \dots, n_g\}, \\ q_y(\boldsymbol{\mu}) &\leq 0, & y &\in \{1, 2, \dots, n_q\}, \\ -\boldsymbol{\mu} &\leq \mathbf{0}, \end{aligned} \quad (4.31)$$

with

$$\begin{aligned} T_{\text{tot}}(\boldsymbol{\mu}) &= \sum_{m=1}^{n_1} T_{1,m}(\boldsymbol{\mu}_{1,m}) + \sum_{m=1}^{n_c} T_{c,m}(\boldsymbol{\mu}_{c,m}), \\ E_{\text{tot}}(\boldsymbol{\mu}) &= \sum_{m=1}^{n_1} E_{1,m}(\boldsymbol{\mu}_{1,m}) + \sum_{m=1}^{n_c} E_{c,m}(\boldsymbol{\mu}_{c,m}), \end{aligned} \quad (4.32)$$

where T_{tot} and E_{tot} are the cycle time and total energy consumption, respectively. $\boldsymbol{\mu}$ is the optimization parameter vector that consists of variables that describe each segment in the trajectory. n_1 and n_c are the number of linear and corner segments, respectively. g_r s are equality constraints consisting of \mathbb{C}^2 continuity conditions (i.e., (4.18) for the case of KCSIA or (4.27) for the case of KCSUA) and geometric bounds that ensure the resulting trajectory coincides with the predefined geometry. q_y s are inequality constraints that restrict the trajectory jerk, acceleration and velocity within kinematic limits. (4.30) is formulated for cases of KCSIA and KCSUA. For the KCSIA case, $T_{c,m}$ and $E_{c,m}$ are obtained from (4.20) and (4.21), respectively. (4.18) are the equality constraints. Conversely, $T_{c,m}$ and $E_{c,m}$ are obtained from (4.29) and (4.21), respectively, with (4.27) as the equality constraints for the KCSUA case. For a PTP case, (4.32) is reduced to

$$T_{\text{tot}}(\boldsymbol{\mu}) = \sum_{m=1}^{n_1} T_{1,m}(\boldsymbol{\mu}_{1,m}), \quad E_{\text{tot}}(\boldsymbol{\mu}) = \sum_{m=1}^{n_1} E_{1,m}(\boldsymbol{\mu}_{1,m}), \quad (4.33)$$

where $\boldsymbol{\mu} = [\boldsymbol{\mu}_{1,1}, \boldsymbol{\mu}_{1,2}, \dots, \boldsymbol{\mu}_{1,n_1}]^\top$ with $v_{1,m,k} = 0$ and $a_{1,m,k} = 0$ as equality constraints.

The maximum and minimum of each objective can be derived by separately optimizing the components of (4.30) as follows:

$$\left. \begin{aligned} T_{\min} &= T_{\text{tot}}(\boldsymbol{\mu}_T) \\ E_{\max} &= E_{\text{tot}}(\boldsymbol{\mu}_T) \end{aligned} \right\}, \quad \boldsymbol{\mu}_T = \arg \min_{\boldsymbol{\mu}} T_{\text{tot}}(\boldsymbol{\mu}),$$

$$\left. \begin{aligned} T_{\max} &= T_{\text{tot}}(\boldsymbol{\mu}_E) \\ E_{\min} &= E_{\text{tot}}(\boldsymbol{\mu}_E) \end{aligned} \right\}, \quad \boldsymbol{\mu}_E = \arg \min_{\boldsymbol{\mu}} E_{\text{tot}}(\boldsymbol{\mu}),$$
(4.34)

where the time and energy saving potentials are respectively defined as:

$$\Delta T = T_{\max} - T_{\min}, \quad \Delta E = E_{\max} - E_{\min}. \quad (4.35)$$

The normalized form of each objective

$$\tilde{T}_{\text{tot}}(\boldsymbol{\mu}) = \frac{T_{\text{tot}}(\boldsymbol{\mu}) - T_{\min}}{\Delta T}, \quad \tilde{E}_{\text{tot}}(\boldsymbol{\mu}) = \frac{E_{\text{tot}}(\boldsymbol{\mu}) - E_{\min}}{\Delta E}, \quad (4.36)$$

is used to map a Pareto frontier, in an objective space, into a normalized objective space such that the anchor points, (T_{\min}, E_{\max}) and (T_{\max}, E_{\min}) , are converted into $\mathbf{r}_T(0,1)$ and $\mathbf{r}_E(1,0)$, respectively. In a similar manner to [55], the optimization problem in (4.30) is formulated:

$$\min_{\boldsymbol{\mu}} \tilde{E}_{\text{tot}}(\boldsymbol{\mu}), \quad (4.37)$$

subject to (4.31) and:

$$\mathbf{N}^\top [\boldsymbol{\psi} - \mathbf{r}] \leq 0, \quad (4.38)$$

with

$$\begin{aligned}
 \mathbf{N} &= \mathbf{r}_E - \mathbf{r}_T = [1, -1]^\top, \\
 \boldsymbol{\psi} &= \left[\tilde{T}_{\text{tot}}(\boldsymbol{\mu}), \tilde{E}_{\text{tot}}(\boldsymbol{\mu}) \right]^\top, \\
 \mathbf{r} &= (1 - \zeta) \mathbf{r}_T + \zeta \mathbf{r}_E, \\
 0 &\leq \zeta \leq 1,
 \end{aligned} \tag{4.39}$$

where $\boldsymbol{\psi}$ is an optimal point that corresponds to a weighting factor ζ . The time and energy optimal cases (i.e., respectively \mathbf{r}_T and \mathbf{r}_E) are represented by $\zeta = 0$ and $\zeta = 1$, respectively.

4.4.2 Optimization constraints

Without loss of generality, it is assumed that a closed path (i.e., a path that has the same starting and ending point) without a smoothed corner at its starting/ending vertex is required. Hence, $n_c = n_l - 1$. It is also assumed that the limits in jerk, acceleration and velocity are symmetrical for each axis. The jerk, acceleration and velocity of a trajectory along the m^{th} linear segment are bounded by their respective kinematic limits $j_{\text{lim},k}$, $a_{\text{lim},k}$, and $v_{\text{lim},k}$ as follows:

$$\left. \begin{aligned}
 j_{\text{max},m,k}^2 - j_{\text{lim},k}^2 &= 0 \\
 a_{1,a,m,k}^2 - a_{\text{lim},k}^2 &\leq 0 \\
 a_{1,d,m,k}^2 - a_{\text{lim},k}^2 &\leq 0 \\
 v_{\text{con},m,k}^2 - v_{\text{lim},k}^2 &\leq 0
 \end{aligned} \right\}, \quad m \in \{1, 2, \dots, n_l\}, \tag{4.40}$$

where $a_{1,a,m,k}$ and $a_{1,d,m,k}$ are respectively the maximum k^{th} acceleration and deceleration along the m^{th} linear segment. A quadratic form is used in (4.40) to ensure differentiability for $\forall \boldsymbol{\mu} \geq 0$. Similarly, the KCSIA cornering trajectory of the m^{th} smoothed corner is restricted by

$$\left. \begin{aligned}
 j_{c,m,k}^2 - j_{\text{lim},k}^2 &\leq 0 \\
 a_{1,c,m,k}^2 - a_{\text{lim},k}^2 &\leq 0 \\
 v_{s,c,m,k}^2 - v_{\text{lim},k}^2 &\leq 0 \\
 v_{e,c,m,k}^2 - v_{\text{lim},k}^2 &\leq 0
 \end{aligned} \right\}, \quad m \in \{1, 2, \dots, n_c\}, \tag{4.41}$$

while the KCSUA cornering trajectory is constricted by

$$\left. \begin{aligned} j_{c,m,k}^2 - j_{\text{lim},m,k}^2 &\leq 0 \\ a_{s,c,m,k}^2 - a_{\text{lim},k}^2 &\leq 0 \\ a_{e,c,m,k}^2 - a_{\text{lim},k}^2 &\leq 0 \\ v_{s,c,m,k}^2 - a_{\text{lim},k}^2 &\leq 0 \\ v_{e,c,m,k}^2 - a_{\text{lim},k}^2 &\leq 0 \end{aligned} \right\}, \quad m \in \{1, 2, \dots, n_c\}. \quad (4.42)$$

Geometric constraints are imposed to ensure that the resulting trajectory coincides with all edges of the predefined geometry. These constraints are described as

$$p_{1,m,k} - p_{0,m,k} + [L_{c,m} - L_m] h_{s,m,k} = 0, \quad m = 1, \quad (4.43)$$

$$p_{1,m,k} - p_{0,m,k} + [L_{c,m-1} + L_{c,m} - L_m] h_{s,m,k} = 0, \quad m \in \{1, 2, \dots, n_1 - 1\}, \quad (4.44)$$

$$p_{1,m,k} - p_{0,m,k} + [L_{c,m-1} - L_m] h_{s,m,k} = 0, \quad m = n_1, \quad (4.45)$$

$$\|\mathbf{p}_{c,m} - \mathbf{p}_{\text{mid},m}\|_2 - \varepsilon_m \leq 0, \quad m \in \{1, 2, \dots, n_1 - 1\}, \quad (4.46)$$

where L_m is the length of the m^{th} linear segment before corner smoothing. Fig. (4.7(a))–(4.7(c)) shows pictorial representations of geometric constraints (4.43)–(4.45). The illustration of (4.46) is shown in Fig. 4.6. For the case of PTP, a geometric constraint

$$p_{1,m,k} - p_{0,m,k} - L_m h_{s,m,k} = 0, \quad m \in \{1, 2, \dots, n_1\}, \quad (4.47)$$

is imposed instead of (4.43)–(4.46).

4.4.3 Pareto generation

The divide and conquer algorithm [36] generates a set of significant Pareto optimal points Ψ (i.e., a Pareto frontier) by recursively taking a weighting factor median ζ of two successive optimal points (i.e., to divide) and then solving (4.37) using the SQP method [9, 33, 62] to produce a new optimal point that is added into Ψ . Weighing factor division is stopped once an insignificant point is obtained, where this point is ignored (i.e., to conquer) and the next

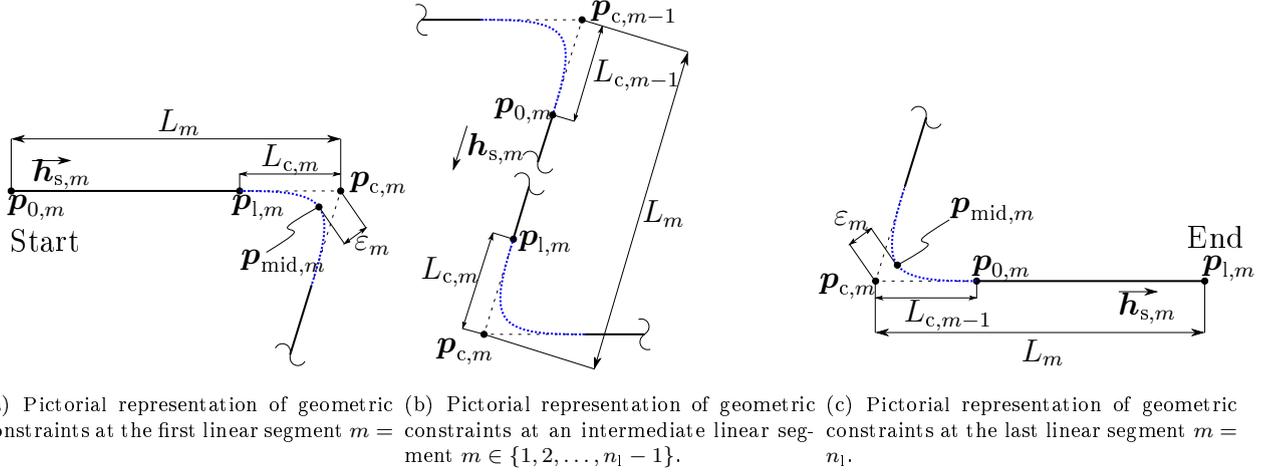


Fig. 4.7: An illustration of geometric constraints (4.43)–(4.45) imposed on the optimization problem (4.37).

unexplored pair of successive optimal points is divided. A significance criterion is set by pre-defining a minimum trade-off level δ_{\min} between consecutive points. The algorithm is initialized by adding \mathbf{r}_T and \mathbf{r}_E into an empty set Ψ . An optimal point is obtained at a median weighting factor $\zeta = 0.5$ and then added into Ψ if it passes the significance criterion. Afterward, the next pair of points in Ψ is explored. The divide and conquer algorithm tends to concentrate Ψ around knee regions of the Pareto frontier since they have higher trade-offs than flat regions. Among these knee region Pareto optimal points, the best trade-off point

$$\boldsymbol{\psi}^* = \arg \min_{\boldsymbol{\psi}} \|\Psi\|_2, \quad (4.48)$$

is defined as the Pareto-optimal point that is nearest to the utopia point (i.e., the origin on the normalized objective space), which corresponds to an optimal solution $\boldsymbol{\mu}^*$ in the decision space. The pseudo-code of the optimization process is illustrated in algorithm 3.

4.4.4 Space complexity analysis

The complexity of solving optimization problems is a fundamental property to consider. Space complexity refers to the amount of memory resource required to perform a computation [34]. Schittkowski describes the space complexity of solving optimization problems by the SQP

Algorithm 3 Trajectory optimization algorithm

```

1: function TRAJECTORYOPTIMIZER( $\mathbf{p}_{cS}, \varepsilon, j_{\text{lim},k}, a_{\text{lim},k}, v_{\text{lim},k}, \delta_{\text{min}}$ )
2:   Calculate anchor points ▷ From (4.34)
3:    $\Psi \leftarrow \{\mathbf{r}_T, \mathbf{r}_E\}$  ▷ Initialize the Pareto set
4:    $\{\zeta_l, \zeta_u\} \leftarrow \{0, 1\}$  ▷ Initialize a pair of weighting factors corresponding to the anchor points
5:    $\mathbf{Z} \leftarrow \{\zeta_l, \zeta_u\}$  ▷ Initialize a weighting factor set
6:   while true do ▷ Divide and conquer algorithm starts here
7:      $\zeta_m \leftarrow 0.5(\zeta_l + \zeta_u)$  ▷ Calculate weighting factor at segment mid-point
8:      $\psi_m \leftarrow$  Solve (4.37) at  $\zeta_m$ 
9:      $\psi_l \leftarrow$  point in  $\Psi$  preceding  $\psi_m$ 
10:     $\psi_u \leftarrow$  point in  $\Psi$  proceeding  $\psi_m$ 
11:     $b_l \leftarrow$  Check  $\{\psi_l, \psi_m\}$  satisfy significance criterion
12:     $b_u \leftarrow$  Check  $\{\psi_u, \psi_m\}$  satisfy significance criterion ▷  $b_l$  and  $b_u$  are boolean variables
13:    if  $b_l \wedge b_u$  then ▷  $\psi_m$  is significant
14:       $\Psi \leftarrow \{\Psi, \psi_m\}$ 
15:       $\mathbf{Z} \leftarrow \{\mathbf{Z}, \zeta_m\}$ 
16:       $\Psi \leftarrow \text{Sort}(\Psi)$  ▷ Sort in ascending order of  $\zeta$ 
17:       $\mathbf{Z} \leftarrow \text{Sort}(\mathbf{Z})$ 
18:       $\zeta_u \leftarrow \zeta_m$ 
19:    else ▷  $\psi_m$  is insignificant
20:       $\zeta_l \leftarrow \zeta_u$ 
21:      if  $\zeta_l < 1$  then
22:         $\zeta_u \leftarrow$  the first  $\zeta > \zeta_u$  in  $\mathbf{Z}$ 
23:      else
24:        break ▷ Terminates the search for  $\psi_m$ 
25:      end if
26:    end if
27:  end while ▷ Divide and conquer algorithms ends here
28:   $\psi^* \leftarrow$  Compute(4.48)
29:   $\mu^* \leftarrow$  map  $\psi^*$  in the decision space
30:  return  $(\Psi, \psi^*, \mu^*)$ 
31: end function

```

method, where an optimization problem with n_{var} variables and n_{con} constraints requires a real working array of length $\mathcal{O}(n_{\text{var}}^2 + n_{\text{var}}n_{\text{con}})$ [70].

The space complexity of the major step in Algorithm 3 (i.e., solving (4.37)) is considered. The number of variables is given by

$$n_{\text{var}} = n_l \dim(\mu_l) + n_c \dim(\mu_c), \quad (4.49)$$

where $\dim(\cdot)$ is the notation for the dimension of a vector. For the KCSIA case, $n_{\text{var}} = 10n_l - 3$ with reference to (4.9) and (4.19). From (4.38), (4.40), (4.41) and (4.43)–(4.46), $n_{\text{con}} = 19n_l - 8$. Therefore, KCSIA has a space complexity of $\mathcal{O}(290n_l^2 - 197n_l + 33) \approx \mathcal{O}(n_l^2)$. From (4.9) and (4.28), the KCSUA case has $n_{\text{var}} = 9n_l - 2$. From (4.38), (4.40) and (4.42)–(4.46), $n_{\text{con}} = 21n_l - 10$. Hence, the space complexity for KCSUA is $\mathcal{O}(270n_l^2 - 168n_l + 24) \approx \mathcal{O}(n_l^2)$. For the PTP case, $n_{\text{var}} = 7n_l$ in accordance with (4.9). From (4.38), (4.40) and (4.47), $n_{\text{con}} = 10n_l + 1$. Therefore, PTP has a space complexity of $\mathcal{O}(119n_l^2 + 7n_l) \approx \mathcal{O}(n_l^2)$. Thus, (4.37) is solved with a space complexity of $\mathcal{O}(n_l^2)$ for all cases.

4.5 Results and discussions

The identified energy consumption model in Section 4.2 is used to generate optimal trajectories. The kinematic limits are specified as $j_{\text{lim},k} = 50,000 \text{ mm/s}^3$, $a_{\text{lim},k} = 1000 \text{ mm/s}^2$ and $v_{\text{lim},k} = 80 \text{ mm/s}$. A geometry with piecewise linear contours (Fig. 4.8) is used to generate optimal corner smoothed trajectories at a tolerance of $0 < \varepsilon \leq 200 \mu\text{m}$ at each corner. This geometry is used for verifying the practical effectiveness of the proposed approach due to the fact that it includes acute and obtuse corner angles in both positive and negative motion directions of each axis whose dynamic properties are typically different due to friction and mechanical inaccuracies. The optimization problem in (4.37) is solved by SQP in a MATLAB[®] environment on a laptop computer with core i7 intel processor, 2.50 GHz CPU, 8 GB RAM and Windows 10 operating system, where the normalized objective function, constraints and their respective gradients are analytically derived and fed to the solver. In order to generate Pareto frontiers of the optimization results, a minimum trade-off level $\delta_{\text{min}} = 0.01$ is used in the divide and conquer algorithm.

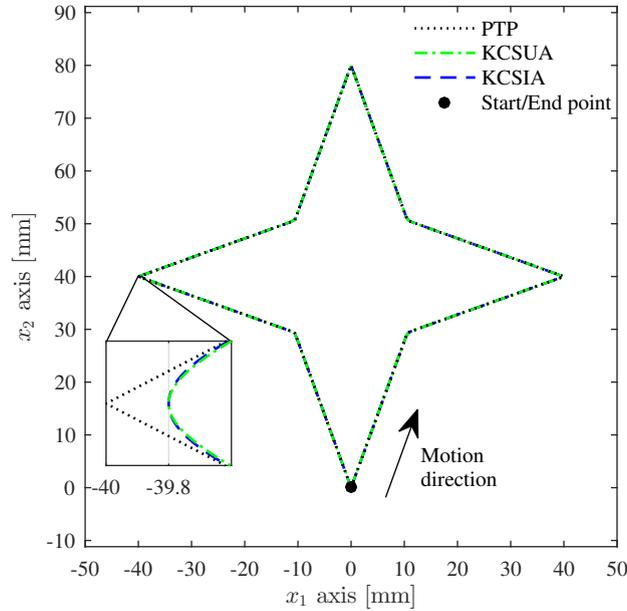


Fig. 4.8: The piecewise linear contours (black dotted lines) used for optimal trajectory generation. An illustration of corner smoothed reference trajectories generated using KCSUA (green dash-dotted lines) and KCSIA (blue dashed lines) are shown.

4.5.1 Optimization results

Fig. 4.9 shows the Pareto frontier of the trajectory optimization results for the cases of KCSIA, KCSUA and PTP. Since the energy consumptions of the KCSIA and KCSUA are lower than those for the PTP for the same cycle time and vice versa, then the Pareto frontiers of KCSIA and KCSUA are better than that for PTP. KCSIA offers the best trade-off solution, where it consumes 2.2 J/cycle less than the overall time-optimal result and it is 1.384 s/cycle faster than the overall energy optimal result. Hence, the best trade-off solution achieves time and energy-saving potentials of $\sim 66\%$ and $\sim 72\%$, which are computed relative to the overall extrema among the three cases. A summary of the best trade-off optimization results is shown in Table 4.2, where KCSIA reduces the cycle time and energy consumption by $\sim 8\%$ and $\sim 3\%$, respectively relative to PTP.

Fig. 4.10 and 4.11 respectively illustrate the jerk, acceleration and velocity profiles of the time, best trade-off and energy optimality scenarios for the x_1 and x_2 axes, where all profiles comply with specified kinematic limits. In all three scenarios, KCSIA has the shortest cycle time while PTP has the longest one. KCSIA and KCSUA have shorter cycle times than PTP since they

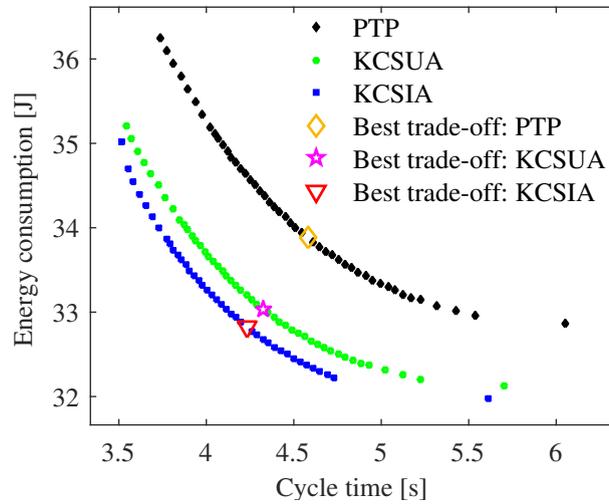


Fig. 4.9: Optimization results represented as a Pareto frontier.

Table 4.2: Summary of optimization results for the best trade-off trajectories

| Trajectory | Saving Potential % | | Reduction relative to PTP % | |
|------------|--------------------|--------------------|-----------------------------|--------------------|
| | Cycle Time | Energy Consumption | Cycle Time | Energy Consumption |
| PTP | ~49 | ~37 | - | - |
| KCSUA | ~62 | ~65 | ~6 | ~3 |
| KCSIA | ~66 | ~72 | ~8 | ~3 |

have non-zero cornering velocities. Since KCSIA achieves higher velocities than KCSUA at corners with obtuse angles, it has a shorter cycle time than KCSUA (see Fig. 4.12). This phenomenon has been described in [84]. Both KCS methods generate cornering motions with relatively low curvature at obtuse corners while the curvature is significantly higher at acute corners, where KCSUA has the highest curvature, as shown in Fig. 4.13. There is no significant curvature change as optimal solutions shift towards energy optimality.

4.5.2 Simulation results

The contouring performance of PTP, KCSUA and KCSIA is simulated using the controller described in Section 2.1.3. The contouring error estimation method in [13] is used in this work. The simulation results are shown in Fig. 4.14. The contouring errors tend to spike at cornering segments since the generated trajectories have non-zero curvatures at corners (Fig.

4.13). KCSIA has a higher contouring error at obtuse corners than KCSUA since it has a higher cornering velocity. As optimal solutions shift toward energy optimality, contouring errors decrease in correlation with decreasing cornering velocity. Contouring errors at sharp corners are higher than those at obtuse corners since curvature is relatively high at these corners. Although KCSUA has a higher curvature than KCSIA, its contouring error decreases significantly as the energy-optimal scenario is approached since it has a lower cornering velocity. Thus, KCSIA has a better trade-off in energy consumption and cycle time, although its contouring performance is worse than KCSUA.

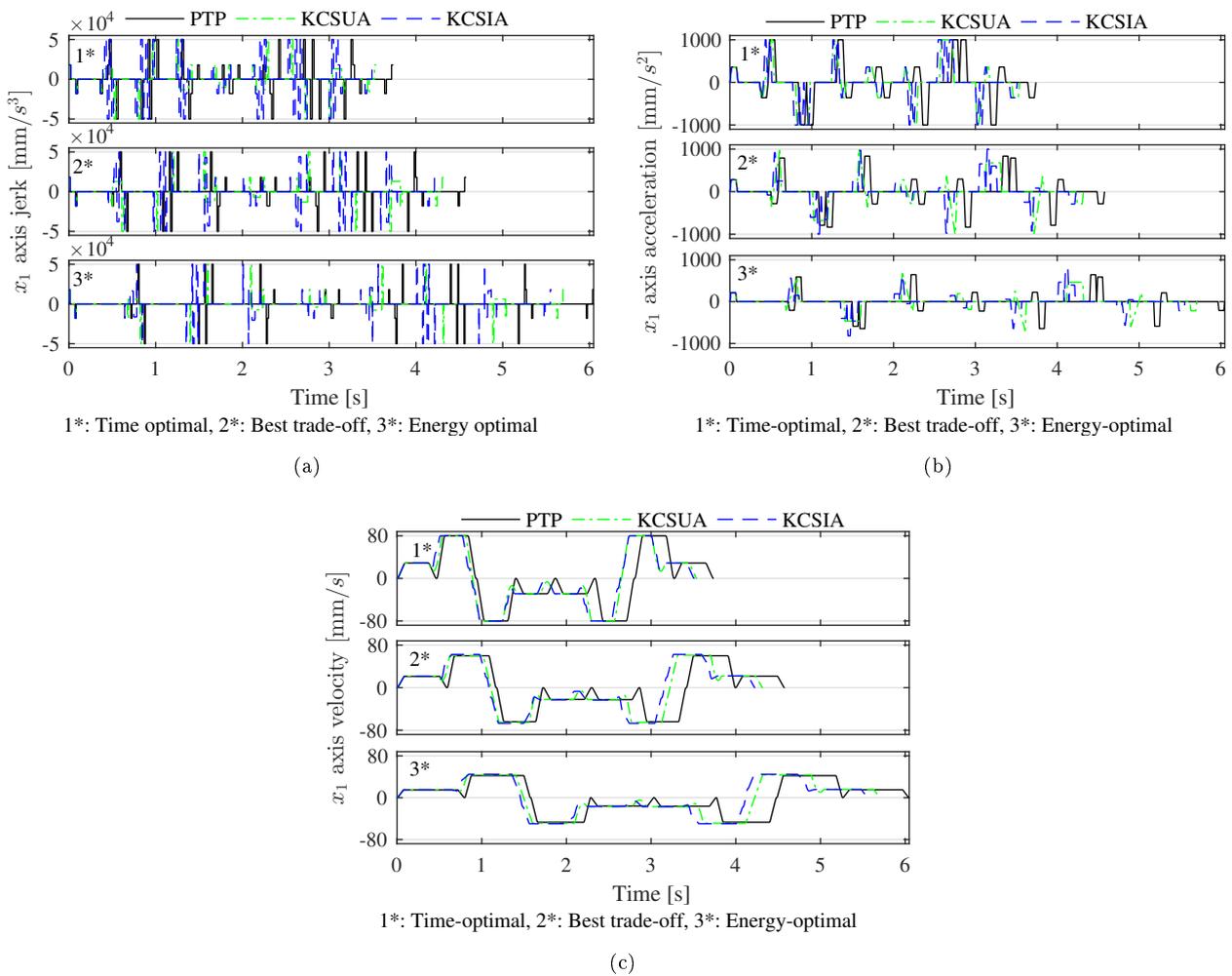


Fig. 4.10: The generated jerk, acceleration and velocity profiles for the x_1 axis.

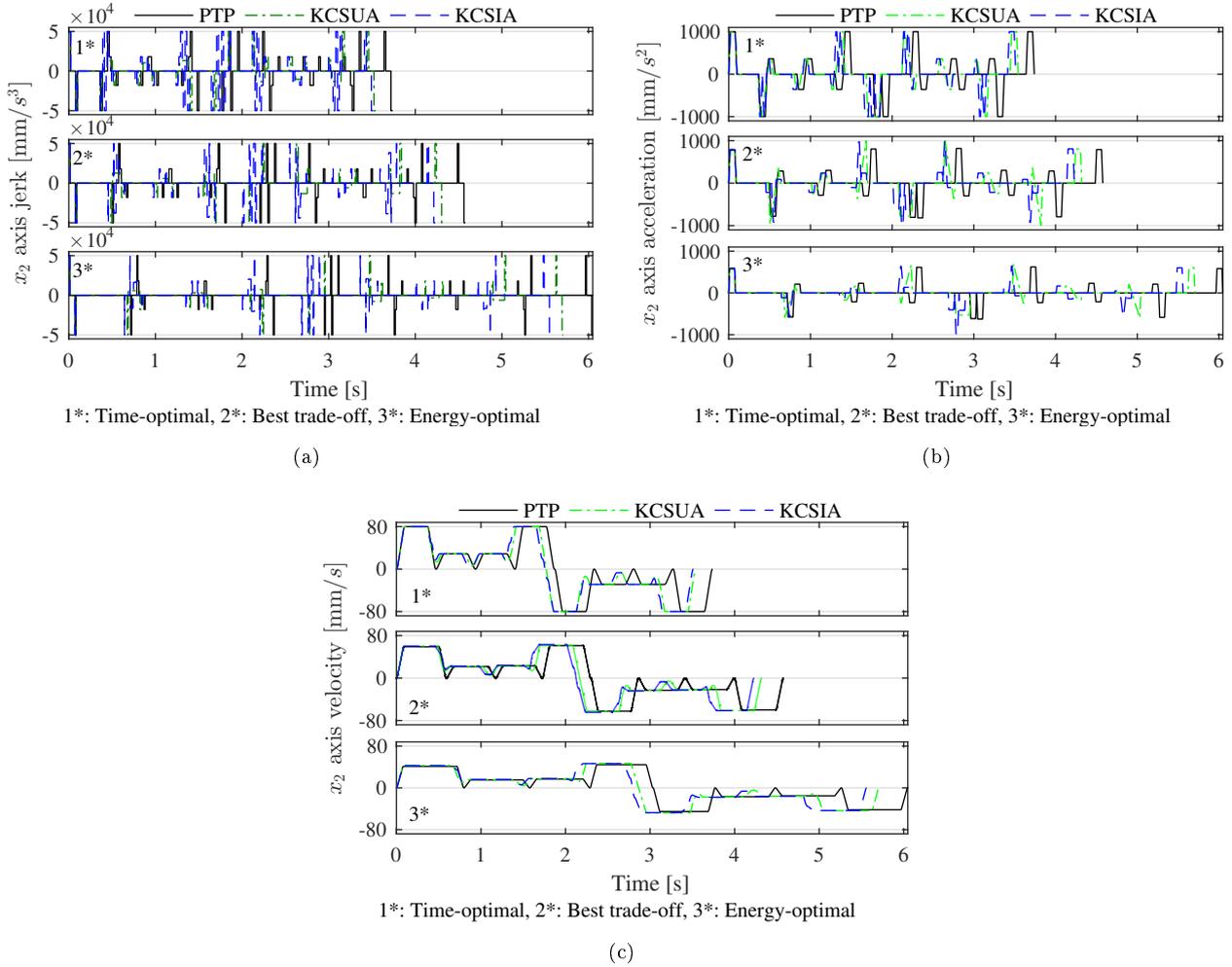


Fig. 4.11: The generated jerk, acceleration and velocity profiles for the x_2 axis.

4.5.3 Experimental results

An experiment is conducted in order to validate the optimization results. Five points corresponding to weighting factors

$$\zeta = \{0, 0.25, \zeta^*, 0.75, 1\}, \quad (4.50)$$

along the Pareto frontier of each case are used to generate trajectories that are implemented on the industrial two-axis machine (Section 2.1.3). Each trajectory is executed through 80 cycles, where five iterations are conducted. The average measured energy consumption per cycle across

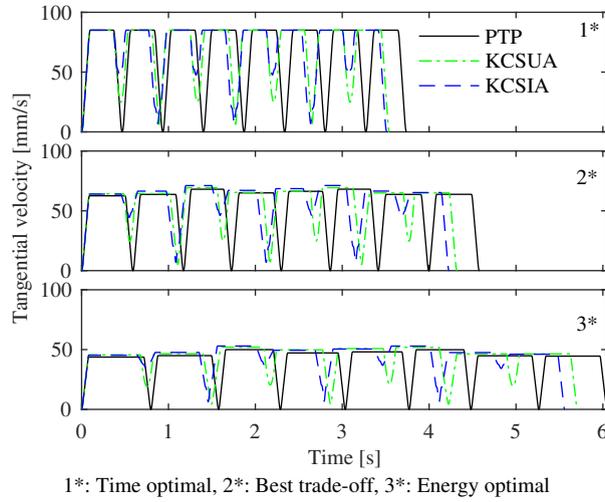


Fig. 4.12: The generated tangential velocities for time, best trade-off and energy optimality scenarios.

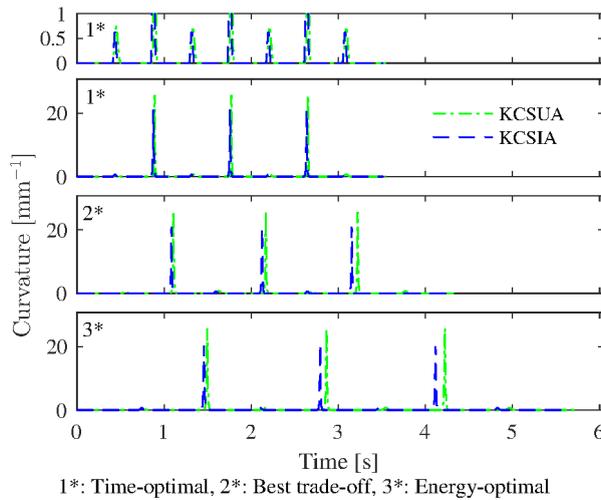


Fig. 4.13: Curvature of the generated trajectories for time, best trade-off and energy optimality scenarios.

all iterations are shown in Fig. 4.15. The experimental results are similar to the optimization ones (see Fig. 4.9), where KCSIA and KCSUA have better energy consumption and cycle time trade-offs than PTP with KCSIA having the best trade-off result. Apart from the time-optimal case (i.e., $\zeta = 0$), KCSIA consumes the least energy for all other weighting factors. The time-optimal trajectory for KCSIA consumes more energy than that for KCSUA due to the fact that it has higher cornering velocity than that for KCSUA (see Fig. 4.12). This results in KCSIA requiring more control effort at the corners. KCSIA provides the overall best trade-off trajectory,

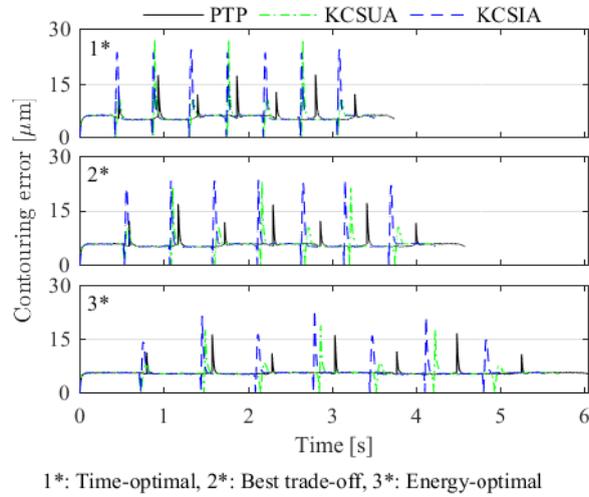


Fig. 4.14: Simulation results of contouring performance.

where its measured energy consumption is 1.7 J/cycle less than the overall time-optimal result. This is equivalent to achieving $\sim 60\%$ of the energy-saving potential computed relative to the overall extrema among the three cases. Experimental results of energy consumption for the best trade-off optimal trajectories are summarized in Table 4.3, where KCSIA reduces energy consumption by $\sim 2\%$ relative to PTP.

The contouring performance of the trajectories for time, best trade-off and energy optimality scenarios is illustrated in Fig. 4.16. Contouring errors are computed by the method described in Section 4.5.2, where instantaneous axial positions are measured using rotary encoders (Section 2.1.3). Time optimal trajectories have the highest contouring error since they have significantly higher velocities. KCSIA has the highest error peaks, located at corners, because it has the highest cornering velocities (see Fig. 4.12). Energy optimal trajectories have the least contouring errors. This is due to fact that they have comparatively lower acceleration and velocity, hence making them easier to be tracked by controllers.

The variation of contouring performance with weighting factor is investigated by generating optimal trajectories for the selected weighting factors in (4.50) and executing each trajectory on the industrial two-axis machine (Section 2.1.3) in a loop for 10 cycles while measuring the contouring error, where the maximum and average errors are recorded. In order to compute the standard deviation of the maximum and mean error values, this procedure is done for a total of five iterations.

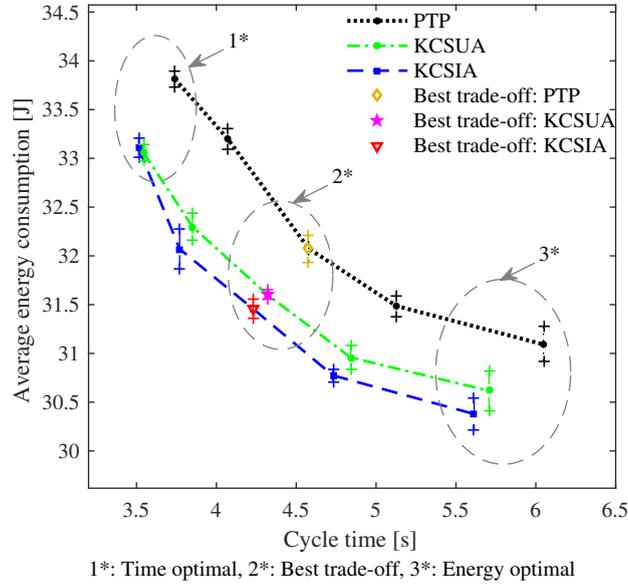


Fig. 4.15: Experimental results represented as a Pareto frontier. The time-optimal, best trade-off and energy-optimal experimental results correspond to the grey-dashed ellipses respectively labeled as 1*, 2* and 3*.

Fig. 4.17 shows the contouring performance results at different weighting factors. The maximum contouring errors are significantly higher than the average ones due to the fact that the instantaneous errors spike at corners as a result of non-zero trajectory curvature as shown in Figs. 4.13, 4.14 and 4.16. The maximum error tends to decrease with an increase in weighting factor, as shown in Fig. 4.17. This is a result of lowering cornering velocity as the energy optimal scenario is approached. The average error also tends to decrease as the weighting factor reaches one. On average, KCSUA tends to have the least contouring error for all weighting factors. It has a lower contouring error than KCSIA as discussed in Section 4.5.1. Although PTP has zero velocity at corners, it has a higher contouring error than KCSUA. This is due to the dominance of non-linear frictional properties such as the Stribeck effect and stick-slip phenomenon in the near-zero velocity region and pre-sliding regime, respectively [45]. Table 4.4 shows a summary of experimental results of contouring performance for the best trade-off optimal trajectories.

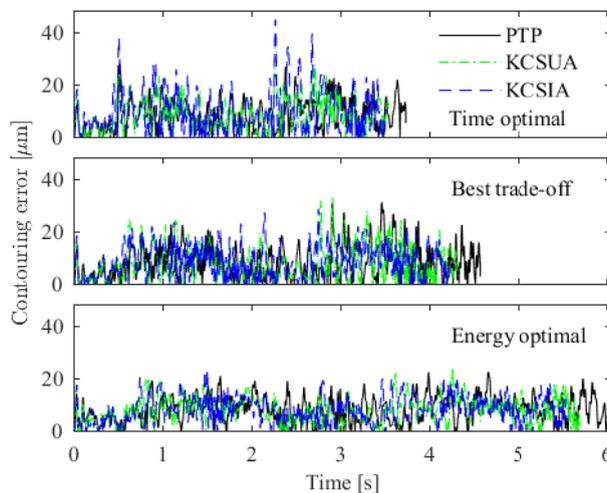


Fig. 4.16: Experimental results of contouring performance.

Table 4.3: Experimental results summary of energy consumption for the best trade-off trajectories

| Trajectory | Energy consumption | | |
|------------|--------------------|--------------------|-----------------------------|
| | Average J | Saving Potential % | Reduction relative to PTP % |
| PTP | 32.1 | ~38 | - |
| KCSUA | 31.6 | ~55 | ~2 |
| KCSIA | 31.5 | ~60 | ~2 |

Table 4.4: Summary of experimental results of contouring performance for the best trade-off trajectories

| Trajectory | Contouring error | | | |
|------------|-----------------------|--------------------|--|--------|
| | Average μm | Max. μm | Reduction relative to PTP Average % | Max. % |
| PTP | 8.9 | 40.5 | - | - |
| KCSUA | 8.3 | 37.4 | ~7 | ~8 |
| KCSIA | 8.9 | 33.0 | - | ~19 |

4.6 Summary

A Pareto-optimal trajectory generation method for piecewise linear contours with smoothed corners is proposed to compromise contradicting objectives of minimizing energy consumption

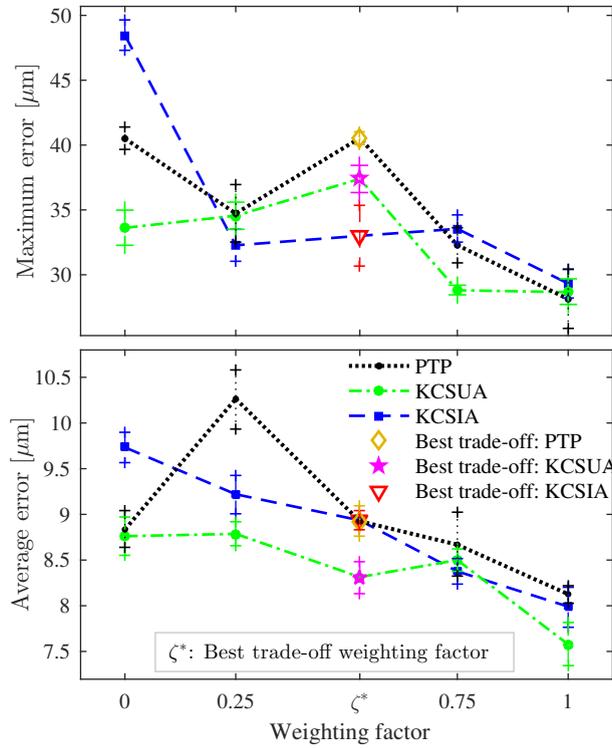


Fig. 4.17: Experimental contouring performance at different weighting factors.

and cycle time in industrial feed drive systems. The linear contours are described by JLAP. KCSIA and KCSUA are used to smoothly blend the motion transition at corners while maintaining a user-specified cornering tolerance. An energy consumption model of an industrial feed drive system is identified by the LSE method and then used in solving the BOOP. Pareto frontiers of three cases: KCSIA, KCSUA and PTP, are generated by the divide and conquer algorithm, where the best trade-off optimal trajectories are obtained.

Experimental results show that the best trade-off trajectory for KCSIA outperforms that for PTP by reducing cycle time and energy consumption by $\sim 8\%$ and $\sim 2\%$, respectively. The KCSIA best trade-off trajectory achieves $\sim 66\%$ and $\sim 60\%$ of the time and energy-saving potentials, respectively computed relative to the overall extrema among the three cases. In terms of contouring performance of best trade-off trajectories, while KCSUA has a $\sim 7\%$ average error reduction, KCSIA has an $\sim 19\%$ maximum error reduction by relative to PTP.

Chapter 5

Pareto Optimization of Cycle Time and Motion Accuracy in Trajectory Planning for Industrial Feed Drive Systems

5.1 Introduction

Manufacturing industries typically use CNC machine tools due to their accuracy, repeatability and speed in performing tasks [95]. Feed drive systems actuate CNC machine tools' motion axes [3]. Ongoing demands for higher production quantity and quality drive researches in motion accuracy improvement and cycle time reduction. Several studies have proposed feedback control structures to reduce tracking [90] and contouring [14, 60] errors in feed drive systems. Chen and Sun propose a nonlinear controller for underactuated systems [17]. A Feedback control strategy is proposed in [18] for regulating ship yaw and roll perturbations in 5-DOF offshore cranes. Such proposals are limited by the accessibility of in-service feed drive system feedback controllers.

Trajectory generation methods have been proposed in the literature to address the objectives of reducing errors and cycle time. Feedforward compensation strategies such as iterative learning control [38, 49] and neural networks and reinforcement learning [50] are proposed for error reduction. A time-optimal trajectory generation approach with consideration to obstacles and

dynamic limits is proposed by Uchiyama et al. for robotic manipulators [93]. Sencer et al. propose time-optimal feed scheduling along B-spline tool paths for 5-axis CNC machine tools [71]. Frequency-optimal acceleration profiles are proposed by Sencer and Tajima for vibration suppression [75]. Kucuk proposes minimum time trajectory generation using cubic spline and 7th order polynomial interpolations [48]. 15-phase sinusoidal jerk profiles are proposed by Fang et al. for cycle time reduction and high-frequency harmonic suppression [31]. Wang et al. propose time-optimal S-curve velocity profile generation for robotic arms [99]. In economic lot scheduling of supply chains, a sub-optimal cycle time of a process lot can be selected to reduce overall production costs [56, 88]. To this end, Jeong et al. present time-optimal and time-fixed jerk-limited velocity profile generation algorithms [44]. Besset and Béarée propose online finite impulse response-based trajectory generation for time-optimal, fixed time and jerk-time fixed cases [7].

Before a contour is fed to FDSs, CAD/CAM systems normally discretize it into a set of linear and circular arc segments using G01 and G02/ G03 commands, respectively. In order to improve machining quality and speed, corner smoothing methods have been proposed in the literature. Corner smoothing methods can be classified as global and local corner smoothing methods based on the span of a fitted curve. Global corner smoothing methods fit a single curve spanning across all segments while local corner smoothing approaches fit a curve between each pair of adjacent segments [98]. Global corner smoothing is normally used for motion planning along short-segment paths [81, 85, 101]. However, it is more difficult to evaluate and constrain the smoothing error for global corner smoothing compared to the local one [96].

Regarding kinematic local corner smoothing, kinematic corner smoothing with interrupted acceleration (KCSIA) has been proposed for cycle time optimality [84]. An energy-time trade-off using KCSIA is studied in [63]. Finite impulse response-based kinematic local corner smoothing have been proposed for reducing time and vibrations [73, 74]. Kinematic local corner smoothing approaches using clothoids [42, 76, 100] and asymmetrical double constant-jerk cornering profiles [98] have been proposed for improving cycle time. Regarding CNC machining, there are several different cases of requirements for surface quality and cycle time: high-accuracy case, high-speed case with a certain accuracy level and a time-fixed case. With the aim of reducing cycle time while satisfying accuracy requirements for piecewise toolpaths, existing local corner smoothing algorithms typically maximize cornering velocities, where cornering errors are driven

to the upper bounds of the accuracy constraints [30, 72]. Under sub-optimal cycle time scenarios, it is no longer necessary to drive these errors to their upper limits. Thus, various trade-off solutions for cycle time and cornering error can be selected by a decision-maker.

Local corner smoothing can be categorized as geometric and kinematic Local corner smoothing, where geometric local corner smoothing separately considers geometric and kinematic constraints in two steps while kinematic local corner smoothing directly plans smooth velocity transitions by considering both constraints in one step [28, 72, 97]. In literature, geometric local corner smoothing using B-spline [41, 105, 106], Pythagorean Hodograph [77] and Bézier [72] curves are proposed. Regarding kinematic local corner smoothing, KCSIA has been proposed for cycle time optimality [84]. An energy-time trade-off using KCSIA is studied in [63]. FIR-based kinematic local corner smoothing have been proposed for reducing time and vibrations [73, 74]. Kinematic local corner smoothing approaches using clothoids [42, 76, 100] and asymmetrical double constant-jerk cornering profiles [98] have been proposed for improving cycle time. Regarding CNC machining, there are several different cases of requirements for surface quality and cycle time: high-accuracy case, high-speed case with a certain accuracy level and a time-fixed case. With the aim of reducing cycle time while satisfying accuracy requirements for piecewise toolpaths, existing local corner smoothing algorithms typically maximize cornering velocities, where cornering errors are driven to the upper bounds of the accuracy constraints [30, 72]. Under sub-optimal cycle time scenarios, it is no longer necessary to drive these errors to their upper limits. Thus, various trade-off solutions for cycle time and cornering error can be selected by a decision-maker.

A Pareto set of a MOOP can be generated by vectorization or scalarization methods [36, 52]. Vectorization methods are stochastic approaches that directly solve the MOOP to produce global-optimal solutions. Their computation cost and stochastic nature limit their application. Scalarization methods solve a MOOP by parameterizing it into a series of SOOPs, resulting in a set of locally optimal solutions. Taking a weighted sum of objectives is a commonly used approach, although it has drawbacks in obtaining solution in non-convex Pareto regions [20]. Approaches such as NBI [21], NNC [55, 69] methods have been proposed. The NBI approach and, to a lesser degree, the NNC are prone to generating non-Pareto optimal solutions. Logist and Van Impe propose a criterion for detecting such solutions [53]. The scalarization methods typically distribute the SOOPs evenly, resulting in computation effort being wasted on insignificant Pareto points. Kim and Weck propose a recursive approach for generating a Pareto

frontier based on an adaptive selection of objective weights in a bi-objective problem [47]. This approach requires four user-defined parameters to control the Pareto frontier approximation. Hashem et al. propose the divide and conquer algorithm for the recursive exploration of significant trade-off regions on the Pareto frontier, where one user-defined parameter, named the minimum trade-off level, is required to control the Pareto frontier resolution [36].

In order to account for the different cases of surface quality and cycle time requirements, a Pareto-optimal local corner smoothing method that offers a trade-off between cycle time and motion accuracy is proposed. A decision maker can systematically select Pareto-optimal solutions to address the requirements under consideration. Piecewise linear contours are considered in this study due to their regular occurrence in CAD/CAM systems. The BOOP is formulated by the NNC method since it is known beforehand whether the Pareto frontier does not have non-convex regions. The motion accuracy improvement objective is indirectly represented by the minimization of corner smoothing extent along an entire contour. As part of the optimization problem formulation, it is considered that a smoothed trajectory's linear and cornering segments are respectively described by JLAP [27] and a modified KCSIA. The optimization problem is constrained by kinematic limits, user-defined cornering tolerances and geometric restriction for avoiding path overlaps. The divide and conquer algorithm [36] is used for generating an approximated Pareto frontier since it only requires one user-defined parameter, named a minimum trade-off level, and it is compatible with the NNC method. Each Pareto-optimal solution is computed using SQP [9, 33, 62]. In this study, the Pareto-optimal solution that is nearest to the utopia point is selected as the best trade-off solution. In a time-fixed case, a solution can be obtained without the need to generate the Pareto front, where the motion accuracy improvement objective is minimized subject to a user-specified cycle time and the above-mentioned constraints. Experimental results of the generated Pareto-optimal trajectories demonstrate the effectiveness the proposed approach.

The contributions of this work are summarized as follows:

- A cycle time and motion accuracy trade-off by Pareto-optimal local corner smoothing is proposed.
- The time-optimal solution has a higher motion accuracy and shorter cycle time than KCSIA.

The chapter is organized as follows: Section 5.2 illustrates the design of Pareto-optimal trajectories, Section 5.3 presents the optimization and experimental results. A discussion on the findings and concluding remarks are set forth in Sections ?? and 5.4, respectively.

5.2 Pareto-optimal trajectory design

A method of generating Pareto optimal trajectories that provide a compromise between motion accuracy and cycle time for piecewise linear contours is illustrated in this section. The bi-objective optimization problem and constraint formulations are depicted below.

5.2.1 Problem formulation

Local corner smoothing approaches reduce cycle time by shortening the tool path length and providing non-zero cornering velocities, which consequently deteriorates contouring performance as shown in [30, 72, 85]. Hence, motion accuracy improvement and cycle time reduction are conflicting objectives in trajectory generation for piecewise linear contours.

Local corner smoothing methods generate cornering trajectories for given machine tool kinematic limits (ie., jerk, acceleration and velocity limits) and a user-specified cornering tolerance $0 < \varepsilon \leq \varepsilon_{ub}$ as shown in Fig. 5.1, where \mathbf{p}_c is the original corner point and the cornering error ε is restricted by upper bound ε_{ub} . The cornering path from position vector \mathbf{p}_s to \mathbf{p}_e is symmetrical about line $\overline{\mathbf{p}_c \mathbf{p}_{mid}}$, where the distance between the start/end of the path and the original corner p_c is the cornering Euclidean distance L_c . \mathbf{h}_s and \mathbf{h}_e are the direction vectors at the start and end of the path, respectively. The minimization of ε and reduction of L_c are non-conflicting objectives. Hence, a SOOP

$$\min_{\boldsymbol{\mu}_{c,m}} L_{c,m}(\boldsymbol{\mu}_{c,m}) \quad (5.1)$$

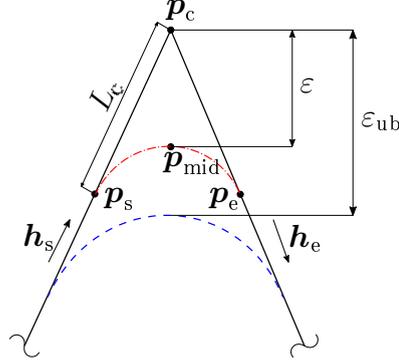


Fig. 5.1: An illustration of a cornering path generated by a corner smoothing method. A corner point \mathbf{p}_c is smoothed by a red dash-dot curve with a cornering error ε bounded by $0 < \varepsilon \leq \varepsilon_{ub}$.

subject to

$$\begin{aligned} \mathbf{g}(\boldsymbol{\mu}_{c,m}) &= \mathbf{0}, \\ \mathbf{q}(\boldsymbol{\mu}_{c,m}) &\leq \mathbf{0}, \\ -\boldsymbol{\mu}_{c,m} &\leq \mathbf{0} \end{aligned} \quad (5.2)$$

is used to describe the smoothing minimization objective for the m^{th} corner, where $\boldsymbol{\mu}_{c,m}$ is parameter vector that consists of variables that describe a cornering trajectory. These variables depend on the corner smoothing method used. \mathbf{g} is an equality constraint vector that imposes motion continuity conditions when stitching the corner path with preceding and succeeding linear segments. \mathbf{q} is an inequality constraint vector that ensures the cornering motion obeys kinematic limits and the user specified cornering tolerance.

The trajectory cycle time

$$T_{\text{cycle}} = \sum_{m=1}^{n_l} T_{l,m}(\boldsymbol{\mu}_{l,m}) + \sum_{m=1}^{n_c} T_{c,m}(\boldsymbol{\mu}_{c,m}) \quad (5.3)$$

is the second objective to be minimized. It is obtained as a sum of the linear and corner segment trajectory durations $T_{l,m}$ s and $T_{c,m}$ s, respectively. $\boldsymbol{\mu}_{l,m}$ is a parameter vector that describes the m^{th} linear segment trajectory. n_l is the number of linear segments. In this study, it is assumed that the end of a piecewise linear contour is not smoothed, hence $n_l = n_c + 1$.

The trade-off between corner smoothing minimization and cycle time reduction is represented as the bi-objective optimization problem

$$\begin{aligned} \min_{\boldsymbol{\mu}} \{L_{c,\text{tot}}(\boldsymbol{\mu}), T_{\text{cycle}}(\boldsymbol{\mu})\} \\ \boldsymbol{\mu} = [\boldsymbol{\mu}_{1,1}, \boldsymbol{\mu}_{c,1}, \boldsymbol{\mu}_{1,2}, \boldsymbol{\mu}_{c,2}, \dots, \boldsymbol{\mu}_{1,n_c}, \boldsymbol{\mu}_{c,n_c}]^\top \end{aligned} \quad (5.4)$$

subject to

$$\begin{aligned} \mathbf{g}(\boldsymbol{\mu}) &= \mathbf{0}, \\ \mathbf{q}(\boldsymbol{\mu}) &\leq \mathbf{0}, \\ -\boldsymbol{\mu} &\leq \mathbf{0} \end{aligned} \quad (5.5)$$

with

$$L_{c,\text{tot}}(\boldsymbol{\mu}) = \sum_{m=1}^{n_c} L_{c,m}(\boldsymbol{\mu}_{c,m}), \quad (5.6)$$

where $L_{c,\text{tot}}$ is the total cornering Euclidean length for corner smoothing. Thus, $L_{c,\text{tot}}$ represents the smoothing objective function for all n_c corners. $\boldsymbol{\mu}$ is the optimization parameter vector consisting of linear and corner segment variables. The \mathbf{g} elements describe geometric and \mathbb{C}^n continuity conditions along a resulting optimal trajectory and \mathbf{q} restricts the trajectory within kinematic limits and user-specified cornering tolerances.

Each objective extremum is obtained by independently minimizing the components in (5.4) as:

$$\begin{aligned} \left. \begin{aligned} L_{c,\text{min}} &= L_{c,\text{tot}}(\boldsymbol{\mu}_{\text{lc}}) \\ T_{\text{max}} &= T_{\text{cycle}}(\boldsymbol{\mu}_{\text{lc}}) \end{aligned} \right\}, \quad \boldsymbol{\mu}_{\text{lc}} = \arg \min_{\boldsymbol{\mu}} L_{c,\text{tot}}(\boldsymbol{\mu}), \\ \left. \begin{aligned} L_{c,\text{max}} &= L_{c,\text{tot}}(\boldsymbol{\mu}_T) \\ T_{\text{min}} &= T_{\text{cycle}}(\boldsymbol{\mu}_T) \end{aligned} \right\}, \quad \boldsymbol{\mu}_T = \arg \min_{\boldsymbol{\mu}} T_{\text{cycle}}(\boldsymbol{\mu}). \end{aligned} \quad (5.7)$$

In a normalized objective space,

$$\begin{aligned}\tilde{L}_{c,\text{tot}}(\boldsymbol{\mu}) &= \frac{L_{c,\text{tot}}(\boldsymbol{\mu}) - L_{c,\text{min}}}{\Delta L_{c,\text{tot}}}, \\ \tilde{T}_{\text{cycle}}(\boldsymbol{\mu}) &= \frac{T_{\text{cycle}}(\boldsymbol{\mu}) - T_{\text{min}}}{\Delta T_{\text{cycle}}}\end{aligned}\tag{5.8}$$

describe the respective individual objectives in (5.4), where $\Delta L_{c,\text{tot}} = L_{c,\text{max}} - L_{c,\text{min}}$ and $\Delta T_{\text{cycle}} = T_{\text{max}} - T_{\text{min}}$ are the saving potentials in corner smoothing and cycle time, respectively. In accordance with the normalized normal constraint method [55], (5.4) is reformulated as:

$$\min_{\boldsymbol{\mu}} \tilde{T}_{\text{cycle}}(\boldsymbol{\mu})\tag{5.9}$$

subject to (5.5) and an additional normalized normal constraint inequality constraint

$$[\boldsymbol{\psi}_{\text{T}} - \boldsymbol{\psi}_{\text{lc}}]^{\top} [\boldsymbol{\psi} - \boldsymbol{\rho}] \leq 0\tag{5.10}$$

that limits the feasible region in the normalized objective space (Fig. 5.2) with

$$\begin{aligned}\boldsymbol{\psi} &= [\tilde{L}_{c,\text{tot}}(\boldsymbol{\mu}), \tilde{T}_{\text{cycle}}(\boldsymbol{\mu})]^{\top}, \\ \boldsymbol{\psi}_{\text{lc}} &= [\tilde{L}_{c,\text{tot}}(\boldsymbol{\mu}_{\text{lc}}), \tilde{T}_{\text{cycle}}(\boldsymbol{\mu}_{\text{lc}})]^{\top}, \\ \boldsymbol{\psi}_{\text{T}} &= [\tilde{L}_{c,\text{tot}}(\boldsymbol{\mu}_{\text{T}}), \tilde{T}_{\text{cycle}}(\boldsymbol{\mu}_{\text{T}})]^{\top}, \\ \boldsymbol{\rho} &= (1 - \zeta) \boldsymbol{\psi}_{\text{lc}} + \zeta \boldsymbol{\psi}_{\text{T}}, \\ 0 &\leq \zeta \leq 1,\end{aligned}\tag{5.11}$$

where a point $\boldsymbol{\psi}$ on a Pareto frontier $\boldsymbol{\Psi}$ corresponds to the point $\boldsymbol{\rho}$ on the CHIM obtained at a weighting factor ζ .

Under a time-fixed case, the corresponding Pareto-optimal solution $\boldsymbol{\psi}$ can be directly obtained without the need for generating $\boldsymbol{\Psi}$ by solving the single objective problem

$$\min_{\boldsymbol{\mu}} L_{c,\text{tot}}(\boldsymbol{\mu})\tag{5.12}$$

Geometric constraints

$$\|\mathbf{p}_{c,m} - \mathbf{p}_{c,m-1}\|_2 - [L_{c,m-1} + s_m + L_{c,m}] = 0 \quad (5.16)$$

are defined in order to avoid overlapping the $(m-1)^{\text{th}}$ and m^{th} cornering paths, where s_m is the path length of the linear segment between the smoothed corners. In order to ensure \mathbb{C}^n smoothness when motion switches from lines to cornering paths and vice versa at time instants t_{ms} , continuity constraints

$$\begin{aligned} \mathbf{r}_1^{(n)}(t_{\text{ms}}) - \mathbf{r}_c^{(n)}(t_{\text{ms}}) &= \mathbf{0}, \\ \mathbf{r}_1^{(n-1)}(t_{\text{ms}}) - \mathbf{r}_c^{(n-1)}(t_{\text{ms}}) &= \mathbf{0}, \\ &\vdots \\ \dot{\mathbf{r}}_1(t_{\text{ms}}) - \dot{\mathbf{r}}_c(t_{\text{ms}}) &= \mathbf{0}, \\ \mathbf{r}_1(t_{\text{ms}}) - \mathbf{r}_c(t_{\text{ms}}) &= \mathbf{0} \end{aligned} \quad (5.17)$$

are established, where the motion profiles for linear and cornering paths are respectively \mathbf{r}_1 and \mathbf{r}_c .

5.2.3 Pareto generation

Ψ is generated using the divide and conquer algorithm recursive structure [36], where (5.9) is solved by SQP [9, 33, 62] for different ζ . In this work, the Pareto optimal point that is nearest to the utopia point is considered to be the best trade-off solution

$$\boldsymbol{\psi}^* = \arg \min_{\boldsymbol{\psi}} \|\Psi\|_2, \quad (5.18)$$

obtained at the weighting factor ζ^* which corresponds to the parameter vector $\boldsymbol{\mu}^*$ in the decision space. Algorithm 4 shows a pseudo code of the optimization process. A candidate solution $\boldsymbol{\psi}_m$ is added into a Pareto set Ψ only if it passes removal [53] and significance criteria.

For an MOOP with m_o objectives, the removal criterion detects non-Pareto regions by checking whether any of the first $m_o - 1$ elements of a permuted Lagrange multiplier vector

$$\hat{\nu}_i = \frac{1}{m_o - 1} \mathbf{E}^{-1} \mathbf{P}^{m_o - i} \mathbf{E} \left[\nu_1, \dots, \nu_{m_o - 1}, \sum_{i=1}^{m_o - 1} \nu_i \right]^\top \quad (5.19)$$

$$i \in \{m_o, m_o - 1, \dots, 1\}$$

are not positive, where $\mathbf{E} = \mathbb{1}_{m_o} - \mathbf{I}_{m_o}$ and ν_i is the i^{th} NNC inequality constraint Lagrange multiplier. \mathbf{P} , $\mathbb{1}_{m_o}$ and \mathbf{I}_{m_o} are $m_o \times m_o$ permutation, all-ones and identity matrices, respectively.

$\boldsymbol{\psi}_m$, with neighboring points $\boldsymbol{\psi}_l = [\psi_{l,1}, \psi_{l,2}]^\top$ and $\boldsymbol{\psi}_u = [\psi_{u,1}, \psi_{u,2}]^\top$, is considered to be a significant Pareto optimal point if

$$\min \left(|\psi_{l,1} - \psi_{m,1}|, |\psi_{l,2} - \psi_{m,2}|, |\psi_{u,1} - \psi_{m,1}|, |\psi_{u,2} - \psi_{m,2}| \right) \geq \delta_{\min}, \quad (5.20)$$

where δ_{\min} is a user-specified minimum trade-off level. Since the removal and significance criteria cannot distinguish between local and global Pareto regions, a Pareto filter [55] is implemented to retain only global Pareto points once the divide and conquer loop is terminated.

5.2.4 Trajectory representation

This section illustrates the motion profiles used for Pareto-optimal trajectory design. With no loss in generality, \mathbb{C}^2 motion continuity is selected in this study, where JLAP and KCSIA respectively define linear and cornering motions.

5.2.4.1 Jerk-limited acceleration profile

JLAPs are \mathbb{C}^2 continuous motion profiles that connect two points by providing acceleration, constant velocity and deceleration phases while obeying boundary conditions and restrictions

on jerk, acceleration and velocity [27, 84]. A JLAP jerk profile

$$\ddot{r}_{1,k}(t) = \begin{cases} j_{\max,k}, & t_{0,l} \leq t < t_{1,a}, \\ 0, & t_{1,a} \leq t < t_{2,a}, \\ -j_{\max,k}, & t_{2,a} \leq t < t_{3,a}, \\ 0, & t_{3,a} \leq t < t_{\text{con}} \\ -j_{\max,k}, & t_{\text{con}} \leq t < t_{1,d}, \\ 0, & t_{1,d} \leq t < t_{2,d}, \\ j_{\max,k}, & t_{2,d} \leq t < t_l \end{cases} \quad (5.21)$$

is defined, where $j_{\max,k}$ is the k^{th} axis maximum jerk magnitude in the motion. The time intervals $t_{0,l} \leq t < t_{3,a}$, $t_{3,a} \leq t < t_{\text{con}}$ and $t_{\text{con}} \leq t < t_l$ are respectively the acceleration, constant velocity and deceleration phases. The total motion duration

$$T_1 = T_{1,a} + T_{2,a} + T_{3,a} + T_{\text{con}} + T_{1,d} + T_{2,d} + T_{3,d} \quad (5.22)$$

is obtained as a sum of time intervals

$$\begin{aligned} T_{1,a} &= t_{1,a} - t_{0,l}, & T_{1,d} &= t_{1,d} - t_{\text{con}}, \\ T_{2,a} &= t_{2,a} - t_{1,a}, & T_{2,d} &= t_{2,d} - t_{1,d}, \\ T_{3,a} &= t_{3,a} - t_{2,a}, & T_{3,d} &= t_l - t_{2,d}, \\ T_{\text{con}} &= t_{\text{con}} - t_{3,a}, \end{aligned} \quad (5.23)$$

that are computed according to kinematic constraints (5.14) and boundary conditions (5.17).

By successive integration of (5.21), the k^{th} axis total displacement

$$\begin{aligned} \Delta r_{1,k} &= v_{s,l,k} T_1 + \frac{1}{2} a_{s,l,k} T_1^2 + j_{\max,k} \left[\frac{1}{6} (T_{1,a} - T_{3,a})^3 - \frac{1}{6} (T_{1,d} - T_{3,d})^3 \right. \\ &\quad + \frac{1}{2} T_{1,a} (T_{1,a} + T_{2,a}) (T_{2,a} + 2T_{3,a}) - \frac{1}{2} T_{1,d} (T_{1,d} + T_{2,d}) (T_{2,d} + 2T_{3,d}) \\ &\quad \left. + (T_{\text{con}} + T_d) \left\{ T_{1,a} (T_{2,a} + T_{3,a}) + \frac{1}{2} (T_{1,a} - T_{3,a}) (T_{1,a} + T_{3,a} + T_{\text{con}} + T_d) \right\} \right] \quad (5.24) \end{aligned}$$

is derived with $T_d = T_{1,d} + T_{2,d} + T_{3,d}$, where the path length $s = \|\Delta \mathbf{r}_1\|_2$. $v_{s,l,k}$ and $a_{s,l,k}$ are the respective k^{th} axis velocity and acceleration components at the linear segment start. Thus, a JLAP can be optimized by describing a parameter vector

$$\boldsymbol{\mu}_1 = [T_{1,a}, T_{2,a}, T_{3,a}, T_{\text{con}}, T_{1,d}, T_{2,d}, T_{3,d}]. \quad (5.25)$$

For JLAPs, kinematic constraints (5.14) are implemented as:

$$\left. \begin{aligned} j_{\text{max},m,k}^2 - j_{\text{lim},k}^2 &= 0 \\ (a_{s,l,k} + j_{\text{max},m,k} T_{1,a,m})^2 - a_{\text{lim},k}^2 &\leq 0 \\ (j_{\text{max},m,k} T_{1,d,m})^2 - a_{\text{lim},k}^2 &\leq 0 \\ v_{\text{con},m,k}^2 - v_{\text{lim},k}^2 &\leq 0 \end{aligned} \right\}, \quad (5.26)$$

$$m \in \{1, 2, \dots, n_1\},$$

with a maximum k^{th} axis velocity

$$v_{\text{con},m,k} = v_{s,l,m,k} + a_{s,l,m,k} (T_{1,a,m} + T_{2,a,m} + T_{3,a,m}) + \frac{1}{2} j_{\text{max},m,k} (T_{1,a,m}^2 - T_{3,a,m}^2) + j_{\text{max},m,k} T_{1,a,m} (T_{2,a,m} + T_{3,a,m}) \quad (5.27)$$

along the m^{th} linear segment.

5.2.4.2 Kinematic corner smoothing with interrupted acceleration

KCSIA is a 2D method of generating near time-optimal \mathbb{C}^2 continuous cornering motions by analytically calculating the cornering velocities, accelerations and durations while enforcing zero acceleration and the same tangential velocities V_c at motion boundaries and obeying user-specified cornering tolerances and kinematic constraints [84]. Based on JLAPs, its jerk profile

$$\ddot{r}_{c,k}(t) = \begin{cases} j_{c,k}, & t_{0,c} \leq t < t_{1,c}, \\ 0, & t_{1,c} \leq t < t_{2,c}, \\ -j_{c,k}, & t_{2,c} \leq t < t_c, \end{cases} \quad (5.28)$$

is defined with

$$\begin{aligned}
 j_{c,k} &= \frac{v_{e,c,k} - v_{s,c,k}}{T_{1,c} (T_{1,c} + T_{2,c})}, & T_{1,c} &= t_{1,c} - t_{0,c} \\
 v_{s,c,k} &= V_c h_{s,k}, & &= t_c - t_{2,c}, \\
 v_{e,c,k} &= V_c h_{e,k}, & T_{2,c} &= t_{2,c} - t_{1,c}. \\
 a_{c,k} &= j_{c,k} T_{1,c},
 \end{aligned} \tag{5.29}$$

$j_{c,k}$ is the cornering jerk. By successive integration of (5.28), a cornering path

$$r_{c,k}(t) = \begin{cases} p_{s,k} + v_{s,c,k} (t - t_{0,c}) + \frac{1}{6} j_{c,k} (t - t_{0,c})^3, & t_{0,c} \leq t < t_{1,c}, \\ p_{1,k} + v_{s,c,k} (t - t_{1,c}) \\ + \frac{1}{2} j_{c,k} T_{1,c} (t - t_{1,c}) (T_{1,c} + t - t_{1,c}), & t_{1,c} \leq t < t_{2,c}, \\ p_{2,k} + v_{s,c,k} (t - t_{2,c}) \\ + \frac{1}{2} j_{c,k} T_{1,c} (t - t_{2,c}) (T_{1,c} + 2T_{2,c} + t - t_{2,c}) \\ - \frac{1}{6} j_{c,k} (t - t_{2,c})^3 & t_{2,c} \leq t < t_c \end{cases} \tag{5.30}$$

is defined with

$$\begin{aligned}
 p_{1,k} &= p_{s,k} + v_{s,c,k} T_{1,c} + \frac{1}{6} j_{c,k} T_{1,c}^3, \\
 p_{2,k} &= p_{1,k} + v_{s,c,k} T_{2,c} + \frac{1}{2} j_{c,k} T_{1,c} T_{2,c} (T_{1,c} + T_{2,c}),
 \end{aligned} \tag{5.31}$$

where $p_{s,k}$ is the k^{th} axis coordinate of the cornering path starting point. The k^{th} axis total displacement while traversing the cornering path

$$\Delta r_{c,k} = \left\{ v_{s,c,k} + \frac{1}{2} j_{c,k} T_{1,c} (T_{1,c} + T_{2,c}) \right\} T_c \tag{5.32}$$

is obtained from (5.30), where $T_c = 2T_{1,c} + T_{2,c}$ is the total motion duration. The cornering Euclidean length

$$L_c = \frac{\Delta r_{c,k}}{(h_{s,k} + h_{e,k})} = \frac{1}{2} V_c T_c \quad (5.33)$$

is derived. The position vector elements at the start, middle and end of the cornering path are

$$\begin{aligned} p_{s,k} &= p_{c,k} - L_c h_{s,k}, \\ p_{\text{mid},k} &= p_{s,k} + v_{s,c,k} \left[T_{1,c} + \frac{T_{2,c}}{2} \right] + \frac{1}{2} j_{c,k} T_{1,c} \left(\frac{T_{2,c}}{2} \right)^2 \\ &\quad + j_{c,k} \left[\frac{1}{6} T_{1,c}^3 + \frac{1}{2} T_{1,c}^2 \left(\frac{T_{2,c}}{2} \right) \right] \quad \text{and} \\ p_{e,k} &= p_{c,k} + L_c h_{e,k}, \end{aligned} \quad (5.34)$$

respectively, where the cornering error $\varepsilon = \|\mathbf{p}_c - \mathbf{p}_{\text{mid}}\|_2$ is derived as

$$\varepsilon = \frac{V_c \{T_{1,c}^2 + 3(T_{1,c} + T_{2,c})^2\}}{24(T_{1,c} + T_{2,c})} \|\mathbf{h}_e - \mathbf{h}_s\|_2. \quad (5.35)$$

For the m^{th} corner, KCSIA solves the optimization problem

$$\begin{aligned} \max_{\boldsymbol{\mu}_c} & V_{c,m} \\ \boldsymbol{\mu}_{c,m} &= [V_{c,m}, T_{1,c,m}, T_{2,c,m}] \end{aligned} \quad (5.36)$$

subject to (5.14)–(5.16) and $-\mathbf{u}_{c,m} \leq 0$. This is followed by the generation of the m^{th} cornering path, where the start/end points are obtained from (5.33) - (5.34) and the path is plotted using (5.30). Afterwards, motions along the line segments that connect the cornering paths are stitched using time optimal JLAP.

In order to solve (5.9) or (5.12), (5.35) is incorporated in the cornering constraints (5.15). The kinematic constraints (5.14) are realized in the form of

$$\left. \begin{aligned} j_{c,m,k}^2 - j_{\text{lim},k}^2 &\leq 0 \\ a_{c,m,k}^2 - a_{\text{lim},k}^2 &\leq 0 \\ v_{s,c,m,k}^2 - v_{\text{lim},k}^2 &\leq 0 \\ v_{e,c,m,k}^2 - v_{\text{lim},k}^2 &\leq 0 \end{aligned} \right\}, \quad m \in \{1, 2, \dots, n_c\}. \quad (5.37)$$

The geometric constraints (5.16) are implemented in a decoupled form

$$\begin{aligned} p_{c,m,k} - p_{c,m-1,k} - [L_{c,m-1} + L_{c,m}] h_{s,m,k} \\ - \Delta r_{1,m,k} &= 0, \\ m \in \{1, 2, \dots, n_l\}. \end{aligned} \quad (5.38)$$

\mathbb{C}^2 continuity constraints are established by setting

$$\left. \begin{aligned} \mathbf{a}_{e,l,m} = \ddot{\mathbf{r}}_{c,m}(t_{0,c}) &= \mathbf{0} \\ \mathbf{a}_{s,l,m+1} = \ddot{\mathbf{r}}_{c,m}(t_c) &= \mathbf{0} \\ \mathbf{v}_{e,l,m} - \mathbf{v}_{s,c,m} &= \mathbf{0} \\ \mathbf{v}_{s,l,m+1} - \mathbf{v}_{e,c,m} &= \mathbf{0} \\ \mathbf{p}_{e,l,m} - \mathbf{p}_{s,m} &= \mathbf{0} \\ \mathbf{p}_{s,l,m+1} - \mathbf{p}_{e,m} &= \mathbf{0} \end{aligned} \right\}, m \in \{1, 2, \dots, n_c\} \quad (5.39)$$

where $\mathbf{a}_{e,l,m}$ and $\mathbf{v}_{e,l,m}$ are the acceleration and velocity vectors at the end of the m^{th} linear motion. $\mathbf{p}_{s,l,m}$ and $\mathbf{p}_{e,l,m}$ are the start and end position vectors of the m^{th} linear motion. All solutions of (5.9) or (5.12) are hereafter referred to as KCSIA* solutions. From each solution, a Pareto-optimal trajectory is generated by plotting linear segment and cornering trajectories in succession according to (5.21) and (5.30), where the linear segment and cornering path variables are retrieved from $\boldsymbol{\mu}$.

5.2.4.3 Space complexity analysis

Space complexity is described as the memory resource required to execute a computation [34]. For an optimization problem with n_{var} variables and n_{con} constraints, a typical SQP solver requires a $\mathcal{O}(n_{\text{var}}^2 + n_{\text{var}}n_{\text{con}})$ long double precision working array [70]. KCSIA* consists of

$$n_{\text{var}} = n_l \dim(\mu_l) + n_c \dim(\mu_c), \quad (5.40)$$

variables, where $\dim(\cdot)$ denotes vector dimension. According to (5.25) and (5.36), $n_{\text{var}} = 10n_l - 3$. With reference to (5.15), (5.26) and (5.37)–(5.39), $n_{\text{con}} = 32n_l - 21$. Hence, the KCSIA* memory demand is double precision working array of length $\mathcal{O}(420n_l^2 - 366n_l + 72) \approx \mathcal{O}(n_l^2)$. In contrast, KCSIA has a $\mathcal{O}(1)$ memory demand since it separately optimizes each corner and linear segment.

5.3 Results and discussions

The validity of the proposed method is tested in this section. The validation process constitutes the generation of Pareto-optimal trajectories for a given piecewise linear contour. Subsequently, contouring error performance experiments are conducted for selected optimal trajectories. Experimental results are compared with KCSIA.

5.3.1 Optimization conditions

The experimental system setup in Section. 2.1.3 with kinematic limits: $\mathbf{j}_{\text{lim}} = [50000, 50000]$ mm/s³, $\mathbf{a}_{\text{lim}} = [1000, 1000]$ mm/s² and $\mathbf{v}_{\text{lim}} = [80, 80]$ mm/s is used for verifying the proposed method. The cornering tolerance at each corner is specified as $0 < \varepsilon \leq 200 \mu\text{m}$. An SQP implementation in MATLAB[®] environment is used to solve (5.9) on a laptop computer having the specifications: core i7 intel processor, 2.50 GHz CPU, 8 GB RAM and Windows 10 operating system. The minimum trade-off level $\delta_{\text{min}} = 0.02$ is used in the divide and conquer algorithm.

In order to verify the effectiveness of the proposed method, an experiment is conducted on the industrial biaxial table. KCSIA and $\zeta = \{0, 0.25, \zeta^*, 0.75, 1\}$ KCSIA* motion profiles are

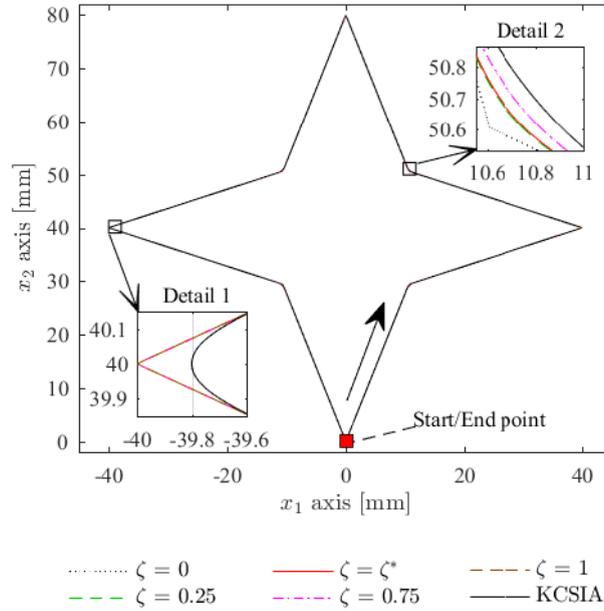


Fig. 5.3: The star-shaped tool path used for Pareto-optimal trajectory generation. The motion direction is shown by the arrow. Details 1 and 2 show the cornering motions at different weighting factors ζ for the acute and obtuse corners, respectively.

fed into the experimental setup as reference trajectories, where 10 trials are conducted to check for the repeatability of results for each trajectory. In this work, the contouring error estimation method in [30] is used, where contouring errors are computed with respect to the original piecewise linear tool paths. In this study, star-shaped (Section 5.3.2) and complex (Section 5.3.3) tool paths are considered for optimal trajectory generation and experimental verification.

5.3.2 Case I: Star-shaped tool path

The star-shaped tool path in Fig. 5.3 is considered for generating Pareto optimal trajectories. This path consists of acute and obtuse corners to check the validity of the proposed method in generating trajectories that trade off cycle time with corner smoothing.

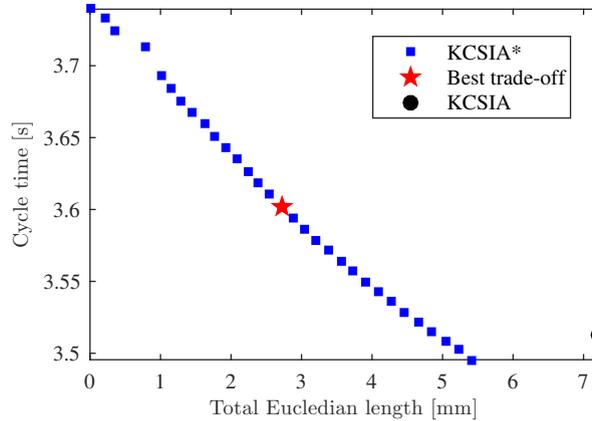


Fig. 5.4: Optimization results of the star-shaped tool path represented as a Pareto frontier.

Table 5.1: Results summary of Pareto-optimal solutions for the star-shaped tool path

| Solution | Weighting factor | Total cornering Euclidean length | Cycle Time | Reduction relative to KCSIA | |
|------------------------------|---------------------|----------------------------------|---------------|-----------------------------|---------------|
| | ζ | $L_{c,tot}$ mm | T_{cycle} s | $L_{c,tot}$ % | T_{cycle} % |
| KCSIA | - | 7.202 | 3.513 | - | - |
| KCSIA* : T_{cycle} optimal | 1 | 5.422 | 3.496 | ~ 25 | ~ 0.5 |
| KCSIA* : $L_{c,tot}$ optimal | 0 | 4.614×10^{-6} | 3.739 | ~ 100 | ~ -6 |
| KCSIA* : Best trade-off | $\zeta^* = 0.53125$ | 2.713 | 3.602 | ~ 62 | ~ -3 |

5.3.2.1 Optimization results

The KCSIA* Pareto frontier is shown in Fig. 5.4, where KCSIA is a dominated solution. The time-optimal KCSIA* solution is a 0.017s (i.e., $\sim 0.5\%$) faster cycle time and 1.780 mm (i.e., $\sim 25\%$) less corner smoothing than KCSIA. KCSIA has an inferior solution since it maximizes cornering velocities, consequently maximizing the cornering Euclidean lengths as shown in (5.33) and (5.36). This results in KCSIA having a reduced cycle time at the cost of a high total cornering Euclidean length (i.e., high cornering errors), while KCSIA* considers both objectives and provides a better performance.

The best trade-off KCSIA* solution is located at $\zeta^* = 0.53125$. It offers 2.709 mm less corner smoothing than the KCSIA* time-optimal result while being 0.137s faster than the minimum corner smoothing result. This corresponds to achieving $\sim 53\%$ and $\sim 53\%$ of the available cycle time and corner smoothing saving potentials. The optimization results are summarized in Table 5.1.

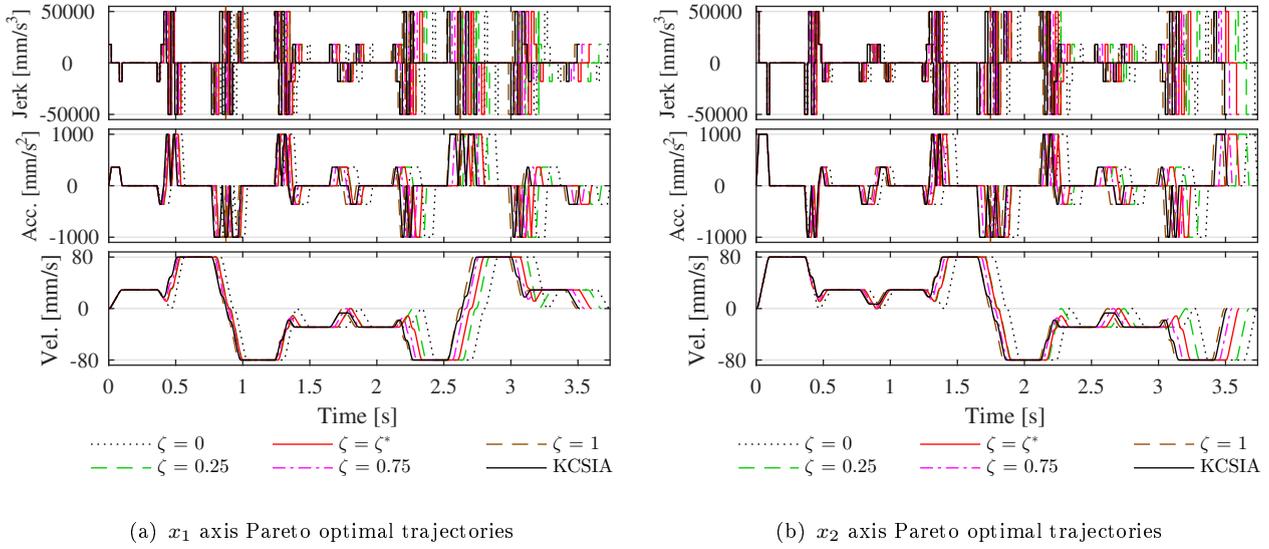


Fig. 5.5: An illustration of Pareto optimal jerk, acceleration and velocity profiles of the star-shaped tool path corresponding to different weighting factors ζ , where kinematic constraints (5.14) are obeyed $\forall \zeta$.

Fig. 5.5 shows the KCSIA* jerk, acceleration and velocity profiles for x_1 and x_2 axes at different ζ values. All profiles are within the pre-defined kinematic constraints (5.14). As $\zeta \rightarrow 1$, $T_{\text{cycle}} \rightarrow T_{\text{min}}$, where at the time-optimal KCSIA* has a shorter T_{cycle} than KCSIA. The KCSIA and KCSIA* tangential velocities are depicted in Fig. 5.6(a). At both acute and obtuse corners, KCSIA has higher cornering velocities than KCSIA* for $\zeta \neq 1$. This is a consequence of (5.36), where the upper bound cornering constraints are activated to maximize velocity as shown in Fig. 5.6(b). For KCSIA*, the cornering velocities and ε_m depend on ζ , where both parameters increase as $\zeta \rightarrow 1$ and vice-versa. The correspondence between ε_m (Fig. 5.6(b)) and the generated cornering paths is shown in details 1 and 2 of Fig. 5.3.

The computation time for generating the Pareto frontier (Fig. 5.4) at $\delta_{\text{min}} = 0.02$ is 188.865 s, where it takes 0.861 s to compute the anchor points (5.7) and 188.865 s to perform the divide and conquer recursions. The average computation time per Pareto-optimal solution is 6.092 s. The choice of δ_{min} affects computation time and the approximation of the best trade-off solution since it is a measure of Pareto frontier resolution. Table 5.2 shows influence of δ_{min} on the best trade-off solution and computation time.

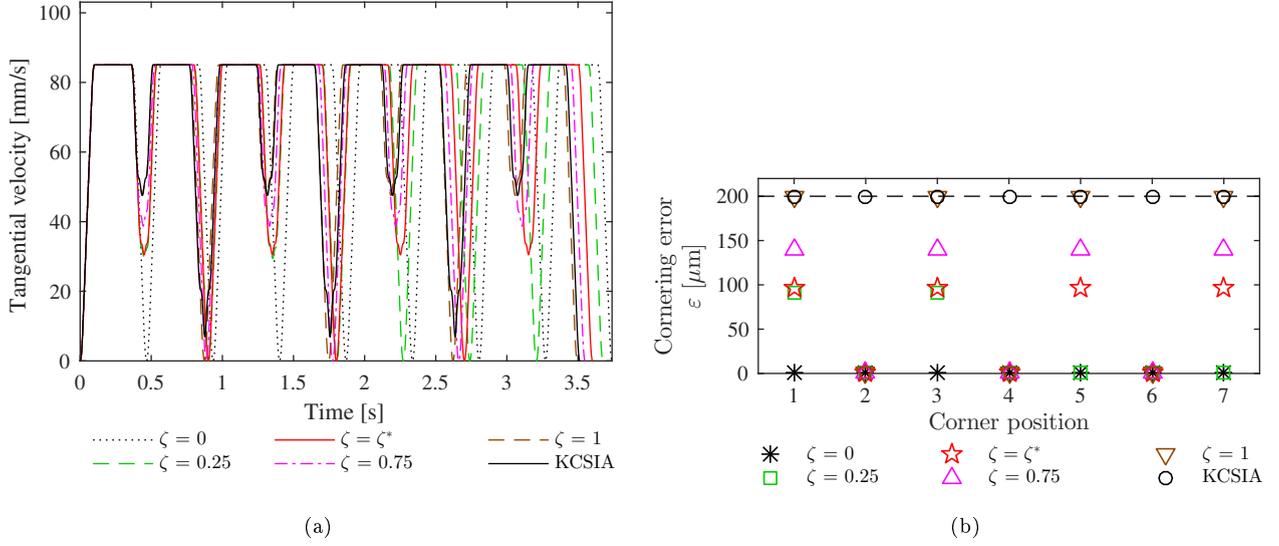


Fig. 5.6: Tangential velocities and cornering errors of Pareto optimal trajectories for the star-shaped tool path at different weighting factors ζ , where ε is within the cornering constraints (5.15).

Table 5.2: Influence the minimum trade-off level on the optimization process

| Min. trade-off level | Best trade-off solution ψ^* | | | Computation time |
|----------------------|----------------------------------|----------------------------------|----------------------|------------------|
| | Weighting factor | Total cornering Euclidean length | Cycle Time | |
| δ_{\min} | ζ^* | $L_{c,\text{tot}}$ mm | T_{cycle} s | s |
| 0.01 | 0.51560 | 2.630 | 3.606 | 306.775 |
| 0.02 | 0.53125 | 2.713 | 3.602 | 188.865 |
| 0.04 | 0.5 | 2.600 | 3.610 | 81.030 |
| 0.08 | 0.5 | 2.600 | 3.610 | 31.015 |

5.3.2.2 Experimental results

KCSIA* has lower error peaks at acute cornering instances than KCSIA and KCSIA* error tends to decrease as $\zeta \rightarrow 0$ at obtuse corners (see Fig. 5.7). The KCSIA* contouring errors are within the pre-defined cornering tolerance (i.e., $0 < \varepsilon \leq 200 \mu\text{m}$). The maximum and average contouring errors of each trajectory for all the trials are illustrated in Fig. 5.8, where it is shown that KCSIA* has a better performance than KCSIA. Since instantaneous contouring errors spike at corners as a result of non-zero trajectory curvature, the maximum errors are

Table 5.3: Experimental results summary of contouring performance for the star-shaped tool path

| Trajectory | Weighting factor ζ | Contouring error | | | |
|------------------------------------|-----------------------------|------------------|---------------|-----------------------------|-----------|
| | | Maximum | Average | Reduction relative to KCSIA | |
| | | μm | μm | Maximum % | Average % |
| KCSIA | - | 198.48 | 14.99 | - | - |
| KCSIA*: T_{cycle} optimal | 1 | 197.27 | 13.86 | ~ 1 | ~ 8 |
| KCSIA*: $L_{c,\text{tot}}$ optimal | 0 | 55.26 | 9.89 | ~ 72 | ~ 34 |
| KCSIA*: Best trade-off | $\zeta^* = 0.53125$ | 104.15 | 11.18 | ~ 48 | ~ 25 |

significantly higher than the average ones (Fig. 5.7). Contouring error tends to increase as cycle time is reduced (i.e., $\zeta \rightarrow 1$). This is attributed to increased cornering velocities (see Fig. 5.6(a)).

The maximum contouring errors for $\zeta = 0$ are non-zero minima $\forall \zeta$ even though the cornering errors ε_m of the reference trajectory are approximately zero (Fig. 5.6(b)). This is the result of vibrations caused by non-linear frictional characteristics such stick-slip and Stribeck effects in pre-sliding regimes and near zero velocity instances, respectively [45].

A 1σ standard deviation is used to validate the contour error results, where Fig. 5.9 shows the consistency of KCSIA* in performing better than KCSIA and the ζ^* KCSIA* trajectory having the best trade-off between cycle time and contouring error. Table 5.3 summarizes the experimental results of contouring performance.

The experimental results indicate that the objectives of reducing cycle time and improving accuracy are contradictory, as illustrated in Fig. 5.9, where the best trade-off solution is selected as the one that is nearest to the unattainable utopia solution. This study also depicts a correlation between corner smoothing and contouring error (Figs. 5.4 and 5.9 in Section 5.3.2, where this result is in agreement with the findings of [30, 63].

Contrary to kinematic local corner smoothing approaches proposed in [42, 73, 74, 84, 86, 98, 100], that achieve near time optimality, where motions at each smooth corner and linear segment are computed separately, the proposed approach generates Pareto-optimal motions for the entire smoothed path. This contrast allows the proposed method to have better results in accuracy and cycle time while KCSIA produces a dominated solution as shown in Figs. 5.4 and 5.9.

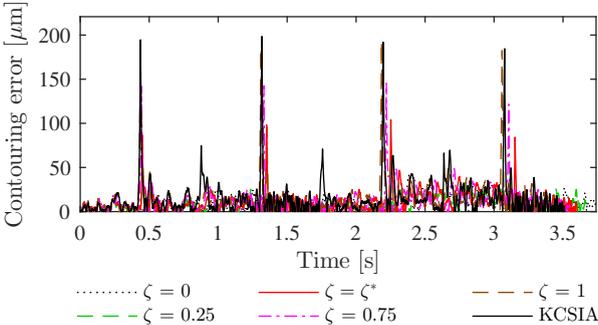


Fig. 5.7: The 7th trial contouring error performance for the star-shaped tool path. The error tends to be within the set cornering tolerance $0 < \varepsilon \leq 200 \mu\text{m}$

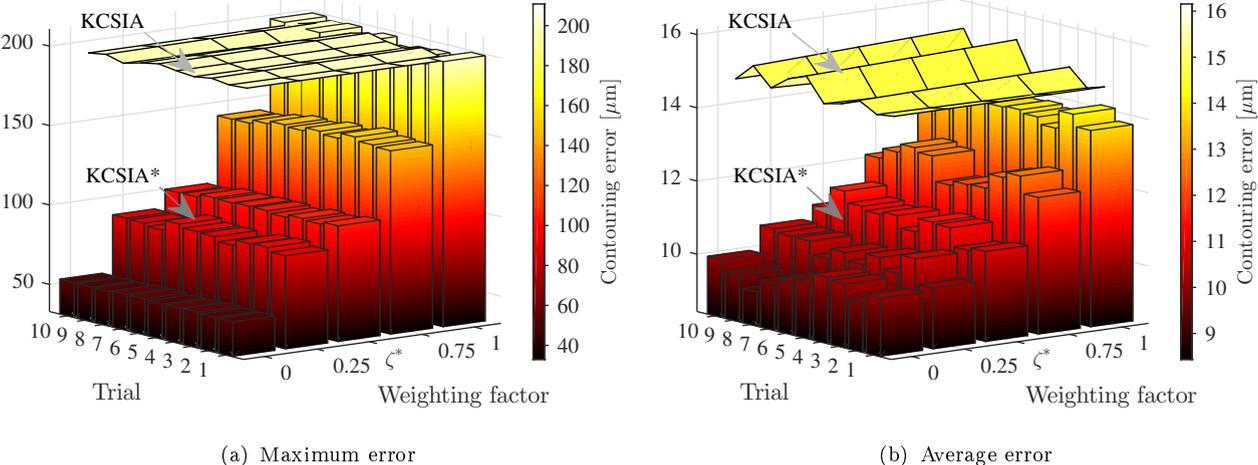


Fig. 5.8: Contouring performance experimental results for the star-shaped tool path.

5.3.3 Case II: Complex tool path

A relatively complicated tool path in the shape of a butterfly (Fig. 5.10) is used to study the effectiveness and limitations of the proposed approach. The path consists of 50 corner points that are interconnected with linear segments having lengths ranging between 0.3 and 6.1 mm.

5.3.3.1 Optimization results

Pareto-optimal solutions of KCSIA* are represented in Fig. 5.11. The KCSIA solution is infeasible since it consists of overlapping cornering paths as shown in detail 2 of Fig. 5.10.

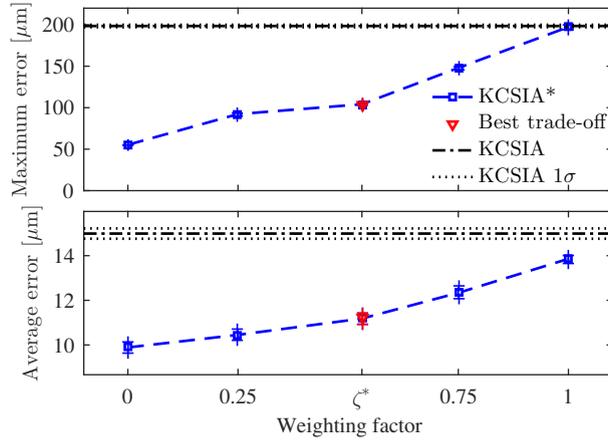


Fig. 5.9: Experimental contouring performance for the star-shaped tool path.

Table 5.4: Results summary of the Pareto optimal solutions for the complex tool path

| Solution | Weighting factor ζ | Total cornering Euclidean length $L_{c,tot}$ mm | Cycle Time T_{cycle} s | Reduction relative to KCSIA* : T_{cycle} optimal | |
|------------------------------|-----------------------------|--|-----------------------------|--|---------------|
| | | | | $L_{c,tot}$ % | T_{cycle} % |
| KCSIA* : T_{cycle} optimal | 1 | 25.383 | 4.580 | - | - |
| KCSIA* : $L_{c,tot}$ optimal | 0 | 1.421×10^{-4} | 6.306 | ~ 100 | ~ -38 |
| KCSIA* : Best trade-off | $\zeta^* = 0.5$ | 10.117 | 5.268 | ~ 40 | ~ -15 |

This is because KCSIA produces cornering motions in isolation from preceding and succeeding corner points. On other hand, as depicted in detail 1 of Fig. 5.10, KCSIA* avoids cornering path overlap by considering the geometric constraints (5.16). The best trade-off solution is located at $\zeta^* = 0.5$, where it reduces corner smoothing by $\sim 40\%$ while it increases cycle time by $\sim 15\%$ relative to the time-optimal KCSIA* solution. A summary of the optimization results is shown in Table 5.4. At $\delta_{min} = 0.02$, the computation time required to generate the Pareto frontier (Fig. 5.11) is 80,011.571 s, where the anchor points are computed in 633.501 s and the divide and conquer recursions are done in 79,378.070 s. The average computation time per Pareto-optimal solution is 2963.392 s.

The generated jerk, acceleration and velocity profiles for different ζ values are within the specified kinematic constraints as shown in Fig. 5.12. The axial velocity constraints are not activated due to the proximity of corner points. Fig. 5.13 shows the extent of corner smoothing at different ζ values. As $\zeta \rightarrow 1$, cornering errors tend to approach the upper bound of the tolerance in order to reduce cycle time.

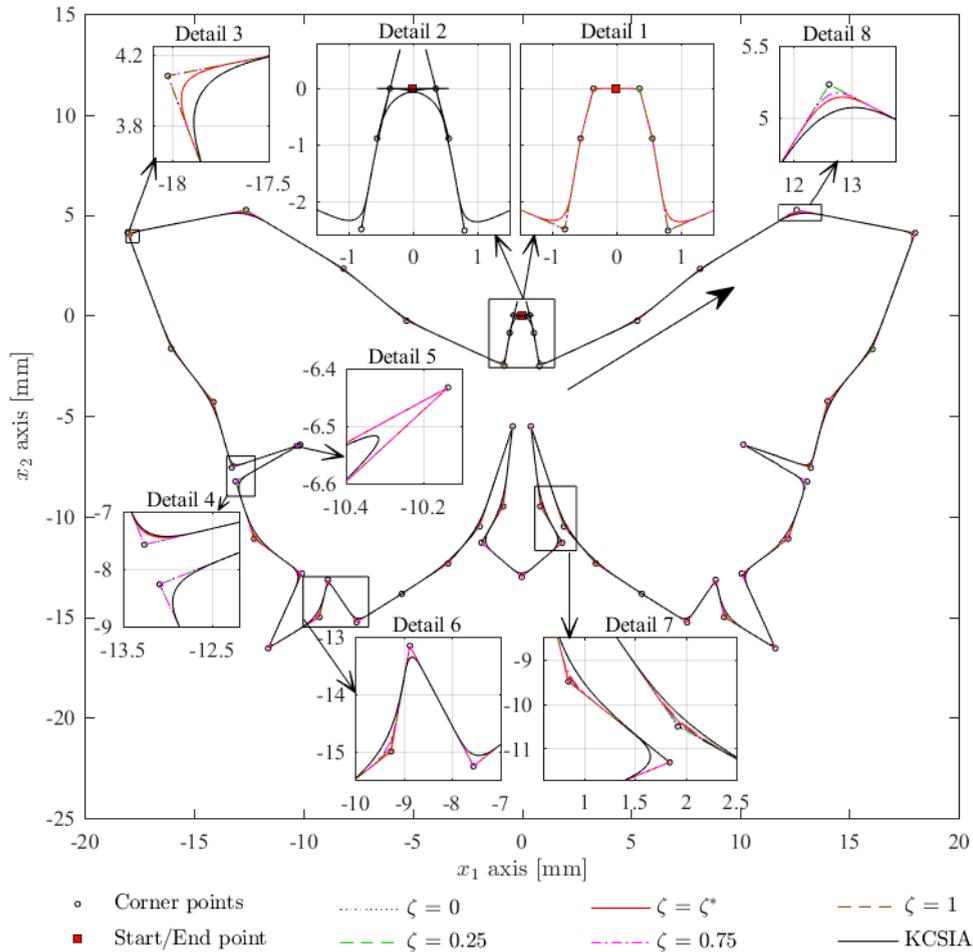


Fig. 5.10: The butterfly-shaped path tool path used to represent complex paths in trajectory generation. The motion direction is shown by the dashed arrow.

5.3.3.2 Experimental results

Contouring error profiles for different ζ values are depicted in Fig. 5.14, where the maximum errors tend to be within the pre-defined cornering tolerance (i.e., $0 < \varepsilon \leq 200 \mu\text{m}$). Fig. 5.15 shows the maximum and average contouring errors of each trial. Similar to Section 5.3.2.2, the maximum contouring errors tend to decrease as $\zeta \rightarrow 0$, where non-zero minima are obtained at $\zeta = 0$. The impact of vibrations at corner points with $\varepsilon_m \approx 0$, by which the motion must stop once and cause a larger tracking error, is shown in Fig. 5.15(b), where the average contouring errors increase as ζ decreases beyond ζ^* . Pareto-optimal trajectories which are generated at

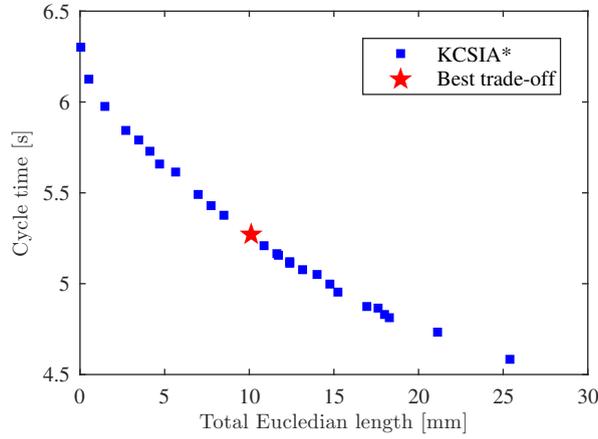


Fig. 5.11: Optimization results for the complex tool path represented as a Pareto frontier.

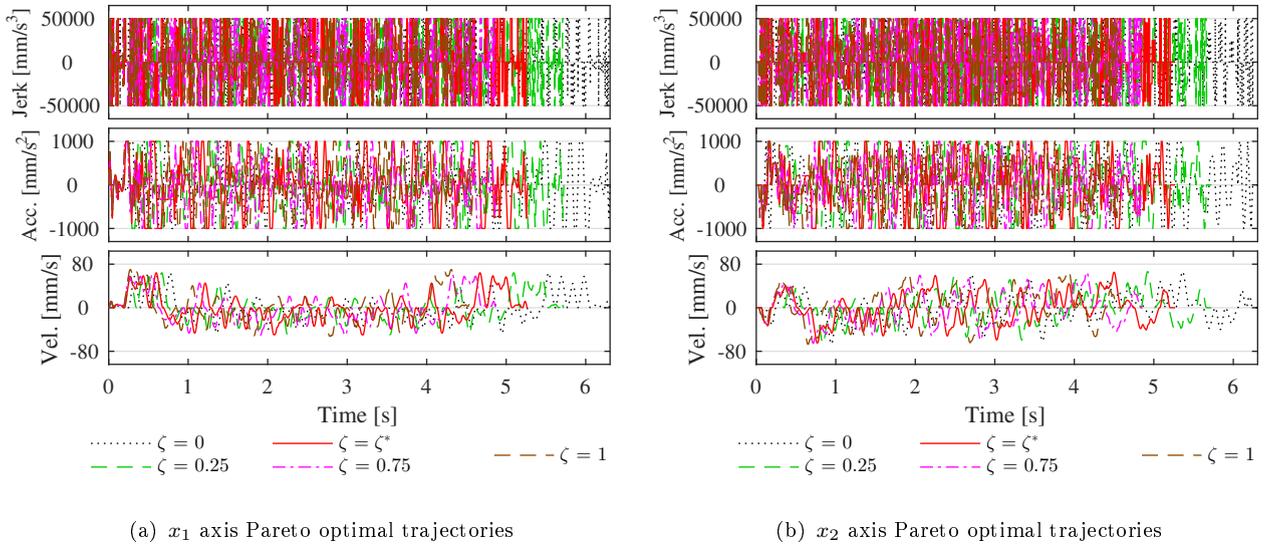


Fig. 5.12: An illustration of Pareto-optimal jerk, acceleration and velocity profiles of the complex tool path corresponding to different weighting factors ζ , where kinematic constraints (5.14) are obeyed $\forall \zeta$.

$\zeta < \zeta^*$ have a higher number of corner points with $\varepsilon_m \approx 0$ than those generated at $\zeta \geq \zeta^*$ (Fig. 5.13).

The repeatability in the maximum and average contouring error results is shown by the 1σ standard deviations in Fig. 5.16. The best trade-off KCSIA* trajectory shows the best compromise between cycle time and maximum/average contouring error. A summary of the experimental results is depicted in Table 5.5.

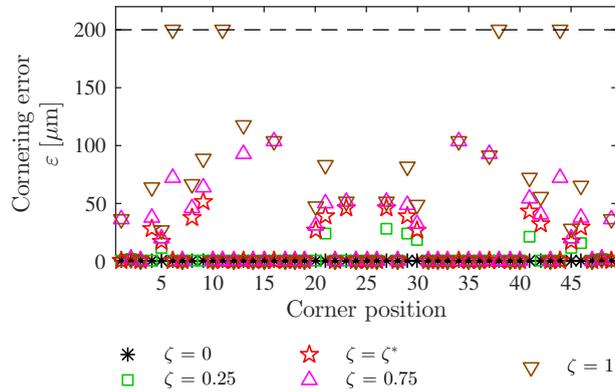


Fig. 5.13: Cornering errors of Pareto-optimal trajectories of the complex tool path at different weighting factors ζ , where ε is within the cornering constraints (5.15).

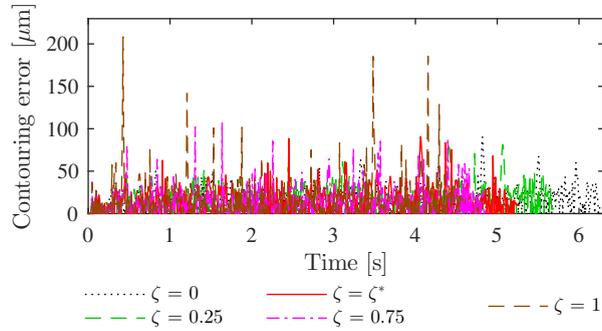


Fig. 5.14: The 3rd trial contouring error performance for the complex tool path. The error tends to be within the set cornering tolerance $0 < \varepsilon \leq 200 \mu\text{m}$

The experiments results indicate that the two objectives are contradictory even for complex tool paths, as illustrated in Fig. 5.16. The correlation between corner smoothing and contouring error is also maintained (Fig. 5.11 and 5.16). The proposed method can also avoid cornering path overlaps while maintaining Pareto-optimality (detail 1 in Fig. 5.10). The computation time of the optimization process is also dependent on the number of linear segments, where it rapidly grows as the number of linear segments increases.

Although significant computation time is required for the complex toolpath in Section 5.3.3, there still exist many mechanical parts consisting of simple and moderate number of linear segments. Hence, the proposed method is effective for such parts.

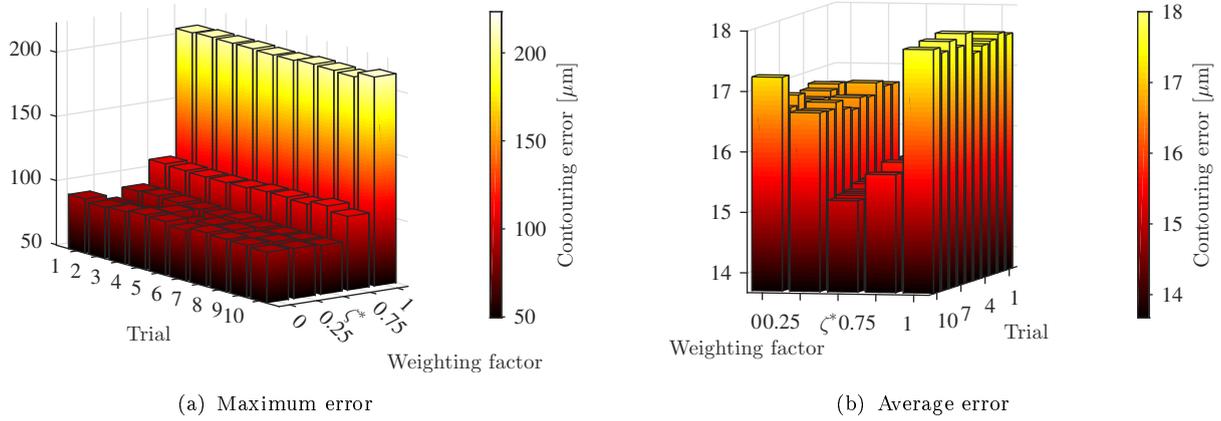


Fig. 5.15: Contouring performance experimental results for the complex tool path.

Table 5.5: Experimental results summary of contouring performance for the complex tool path

| Trajectory | Weighting factor ζ | Contouring error | | | |
|------------------------------------|-----------------------------|--------------------------|--------------------------|--|-----------|
| | | Maximum μm | Average μm | Reduction relative to KCSIA*: T_{cycle} optimal | |
| | | | | Maximum % | Average % |
| KCSIA*: T_{cycle} optimal | 1 | 209.07 | 17.62 | - | - |
| KCSIA*: $L_{\text{c,tot}}$ optimal | 0 | 90.53 | 16.76 | ~ 57 | ~ 5 |
| KCSIA*: Best trade-off | $\zeta^* = 0.53125$ | 89.49 | 15.33 | ~ 57 | ~ 13 |

5.4 Summary

This chapter proposes a method of Pareto-optimal corner smoothing to trade-off between cycle time and motion accuracy for industrial feed drive systems with piecewise linear contours. The total cornering Euclidean length is used as a motion accuracy representative in the BOOP formulated by the NNC approach, where kinematic limits, continuity conditions and a user-specified cornering tolerance are described as constraints. JLAPs describe linear motions, while a modified KCSIA defines smooth corner profiles. A Pareto frontier is generated by the divide and conquer algorithm, where the solution nearest to the utopia point is selected as the best trade-off solution. The proposed method's effectiveness is verified via experiments. Relative to KCSIA, the best trade-off solution reduces the maximum and average contouring errors by $\sim 48\%$ and $\sim 25\%$ while it increases cycle time by $\sim 3\%$.

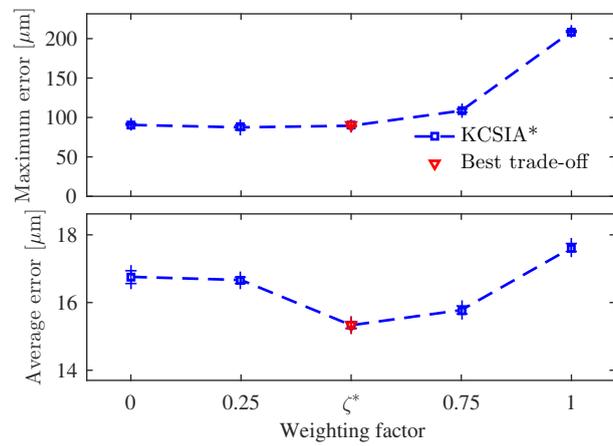


Fig. 5.16: Experimental contouring performance for the complex tool path.

Algorithm 4 Trajectory optimization algorithm

```

1: function TRAJECTORYOPTIMIZER( $\mathbf{p}_{c,s}$ ,  $\varepsilon_{ub}$ ,  $j_{lim,k}$ ,  $a_{lim,k}$ ,  $v_{lim,k}$ ,  $\delta_{min}$ )
2:   compute anchor points ▷ See (5.7)
3:    $\Psi \leftarrow \{\psi_{lc}, \psi_T\}$  ▷ Pareto frontier Initialization
4:    $\psi_l \leftarrow \psi_{lc}$  ▷ Initial lower parent point
5:    $\psi_u \leftarrow \psi_T$  ▷ Initial upper parent point
6:    $\{\zeta_l, \zeta_u\} \leftarrow \{0, 1\}$  ▷ Parent point weighting factor initialization
7:    $\mathbf{Z} \leftarrow \{\zeta_l, \zeta_u\}$  ▷ Weighting factor set initialization
▷ Divide and conquer loop
8:   while true do
9:      $\zeta_m \leftarrow 0.5 (\zeta_l + \zeta_u)$  ▷ Compute weighting factor median
10:     $\psi_m \leftarrow$  solve (5.9) at  $\zeta = \zeta_m$ 
11:     $b_r \leftarrow$  removalCriterion( $\psi_m$ ) ▷ See (5.19)
12:     $b_s \leftarrow$  significanceCriterion( $\psi_m, \psi_l, \psi_u, \delta_{min}$ ) ▷ See (5.20)
13:    if  $b_s$  then ▷  $\psi_m$  is significant
14:      if  $!b_r$  then ▷  $\psi_m$  should not be removed
15:         $\Psi \leftarrow \{\Psi, \psi_m\}$ 
16:         $\mathbf{Z} \leftarrow \{\mathbf{Z}, \zeta_m\}$ 
17:         $\Psi \leftarrow$  sort( $\Psi$ ) ▷ Sort in ascending order of  $\zeta$ 
18:         $\mathbf{Z} \leftarrow$  sort( $\mathbf{Z}$ )
19:      end if
20:       $\psi_u \leftarrow \psi_m$  ▷ Update upper parent point
21:       $\zeta_u \leftarrow \zeta_m$ 
22:    else ▷  $\psi_m$  is insignificant
23:       $\psi_l \leftarrow \psi_u$  ▷ Update lower parent point
24:       $\zeta_l \leftarrow \zeta_u$ 
25:      if  $\zeta_l < 1$  then
26:         $\psi_u \leftarrow$  the point preceding  $\psi_l$  in  $\Psi$ 
27:         $\zeta_u \leftarrow$  the first  $\zeta > \zeta_u$  in  $\mathbf{Z}$ 
28:      else
29:        break ▷ Terminate the divide and conquer loop
30:      end if
31:    end if
32:  end while
33:   $\Psi \leftarrow$  paretoFilter( $\Psi$ ) ▷ From [55]
34:   $\psi^* \leftarrow$  compute (5.18)
35:   $\mu^* \leftarrow$  map  $\psi^*$  in the decision space
36:  return ( $\Psi, \psi^*, \mu^*$ )
37: end function

```

Chapter 6

Conclusions and Future Works

6.1 Conclusions

CNC machine tool technology is faced with the challenge of striving towards conflicting goals of saving energy, cycle time reduction and accuracy improvement. The proposed solutions in the literature fail to address this challenge by either having applicability restrictions or an inability to consider multiple objectives. Thus, the motive of this thesis is to propose solutions to the multi-objective challenge that apply to in-service and future CNC machinery. To this end, Pareto-optimal trajectory generation methods have been proposed. In the order in which they are presented in the thesis, the following conclusions are drawn:

- A trajectory generation method that provides a trade-off between energy consumption and cornering error under user-specified cycle time restrictions is presented in Chapter 3. The proposed method solves the BOOP to produce a Pareto frontier, where each solution corresponds to a trajectory adhering to kinematic limits, cornering, cycle time and \mathbb{C}^2 continuity constraints. A Pareto-optimal trajectory consisting of linear and cornering segments is represented by JLAPs and KCSIA, respectively. The solution that is nearest to the objective utopia is selected as the best compromising trajectory.

- Chapter 4 presents Pareto-optimal corner smoothing of piecewise linear contours to compromise cycle time reduction with saving energy under kinematic, cornering and \mathbb{C}^2 continuity constraints. Motions along linear segments are described by JLAPs. KCSIA and KCSUA are considered for profiling cornering motions. An LSE-based identification approach is presented for modeling the energy consumed by an industrial feed drive system experimental setup. The identified energy consumption model is used in the NNC formulation of the BOOP. Pareto frontiers for KCSIA and KCSUA are generated by the divide and conquer algorithm and compared with a PTP motion case. Simulation and experimental results show that KCSIA and KCSUA Pareto frontiers have a better performance compared to PTP. KCSIA offers the best trade-off solution, where it respectively achieves $\sim 66\%$ and $\sim 60\%$ of the time and energy-saving potentials computed relative to the overall extrema among the three cases.
- A method that offers a trade-off between cycle time and motion accuracy by Pareto-optimization is proposed in Chapter 5. The method uses the total cornering Euclidean length to represent the extent of corner smoothing along an entire piecewise linear contour, where kinematic limits, continuity conditions and a user-specified cornering tolerance are described as constraints. The BOOP formulated by the NNC approach, where a Pareto frontier is generated by the divide and conquer algorithm. Linear and cornering motions are described by JLAPs and KCSIA*. Experimental results validate the Pareto optimality of the generated solutions, where the best trade-off solution reduces the maximum and average contouring errors by $\sim 48\%$ and $\sim 25\%$ while it increases cycle time by $\sim 3\%$.

6.2 Future works

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

- The method in Chapter 3 offers a trade-off between energy consumption and cornering error but assumes identical cornering errors at all corners. Better Pareto-optimal results may be obtained by removing this constraint. In order to achieve this without expanding

the objective space dimensions, the corner smoothing minimization objective used in Chapter 5 can be used.

- In the case of machining operations, cutting forces increase the amount of energy consumption significantly. Cutting force and energy consumption models for machining operations can be acquired using a method similar to [39] and incorporated in formulating a MOOP as an extension to the method in Chapter 4.
- Although the trajectory generation method for trading off between cycle time and energy consumption (Chapter 4) can be applied to nonlinear mechanical systems under the condition that the nonlinearities are identifiable, the LSE-based energy model identification method presented in Section 4.2.1 is limited by the system dynamics (2.1) selected in this thesis. This is due to the fact that the constraints (4.3) used in the identification process are derived from the system dynamics. Thus, considerations for energy consumption model identification of nonlinear mechanical systems and a generalized identification approach are left as future works.
- The trajectory generation method for compromising between cycle time and energy consumption (Chapter 4) can be extended for smooth contours. The trajectory parameter vector $\boldsymbol{\mu}$ in (4.37) can be modified to describe such contours. The formulations of the objective function (4.37) and constraints ((4.31) and (4.38)) remain unchanged regardless of the contour (smooth or piecewise linear). This extension shall be explored as future works.
- The proposed methods are limited to symmetrical line-to-line corner transitions. Asymmetry in corner smoothing increases the degree of freedom in the optimization process, and thus, possibly provides better Pareto-optimal solutions. Consideration for transitions other than line-to-line ones can generalize the proposed approaches.
- As local corner smoothing approaches, the proposed methods may fail to find solutions if the lower bound of cornering tolerance is relatively high compared to the corner point proximity. Pareto optimal global corner smoothing considerations are left as future work.
- The method in Chapter 5 requires significant computational resources for complex tool-paths with many corner points. Computational cost reduction is left as future works.

-
- Since the method in Chapter 5 does not include actual contouring error measurements in the optimization process, the highest achievable cornering accuracy is limited by the feed drive system controller. Feed forward compensation strategies [38, 49, 50] can be incorporated in the optimization process to overcome controller limitations and improve the proposed method's performance. This research avenue will be considered as future works.
 - In this thesis, a fixed controller was used as a representation of inaccessible controllers that commonly occur in industrial CNC machine tools. Under the assumption that a controller is accessible, controller parameters can be optimized together with the trajectory. In future works, controller and trajectory optimization considerations shall be explored.
 - The methods in Chapter 3–5 can be extended to 5-axis CNC machine tool applications by incorporating tool position and orientation tolerances at corners in the BOOPs.
 - The results of Chapters 3–5 confirm that the pairs of objectives are contradictory. Thus, it is possible to the proposed approaches by considering all the objectives in one MOOP formulation, and thus, generating a Pareto surface of solutions.

Bibliography

- [1] Adderley, K. J. (1966). Another model to demonstrate the two wattmeter method of power measurement. *The International Journal of Electrical Engineering & Education*, 4(4):523–526.
- [2] Altintas, Y. and Erkorkmaz, K. (2003). Feedrate optimization for spline interpolation in high speed machine tools. *CIRP Annals*, 52(1):297–302.
- [3] Altintas, Y., Verl, A., Brecher, C., Uriarte, L., and Pritschow, G. (2011). Machine tool feed drives. *CIRP Annals*, 60(2):779–796.
- [4] Antipova, E., Pozo, C., Guillén-Gosálbez, G., Boer, D., Cabeza, L., and Jiménez, L. (2015). On the use of filters to facilitate the post-optimal analysis of the pareto solutions in multi-objective optimization. *Computers & Chemical Engineering*, 74:48–58.
- [5] Aramcharoen, A. and Mativenga, P. T. (2014). Critical factors in energy demand modelling for cnc milling and impact of toolpath strategy. *Journal of Cleaner Production*, 78:63–74.
- [6] Barre, P. J., Bearee, R., Borne, P., and Dumetz, E. (2005). Influence of a jerk controlled movement law on the vibratory behaviour of high-dynamics systems. *Journal of Intelligent and Robotic Systems*, 42(3):275–293.
- [7] Besset, P. and Béarée, R. (2017). Fir filter-based online jerk-constrained trajectory generation. *Control Engineering Practice*, 66:169–180.
- [8] Bi, Z. and Wang, L. (2012). Optimization of machining processes from the perspective of energy consumption: A case study. *Journal of Manufacturing Systems*, 31(4):420–428.
- [9] Boggs, P. T. and Tolle, J. W. (1995). Sequential quadratic programming. *Acta Numerica*, 4:1–51.

-
- [10] Bosetti, P. and Bertolazzi, E. (2014). Feed-rate and trajectory optimization for cnc machine tools. *Robotics and Computer-Integrated Manufacturing*, 30(6):667–677.
- [11] Branke, J., Deb, K., Dierolf, H., and Osswald, M. (2004). Finding knees in multi-objective optimization. In Yao, X., Burke, E. K., Lozano, J. A., Smith, J., Merelo-Guervós, J. J., Bullinaria, J. A., Rowe, J. E., Tiño, P., Kabán, A., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN VIII*, pages 722–731, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [12] Bui, B. D. and Uchiyama, N. (2016). Sliding mode contouring controller design with adaptive friction compensation for three-axis machine tools. In *2016 American Control Conference (ACC)*, pages 2217–2222. Institute of Electrical and Electronics Engineers (IEEE).
- [13] Bui, B. D., Uchiyama, N., and Sano, S. (2015). Nonlinear friction modeling and compensation for precision control of a mechanical feed-drive system. *Sensors and Materials*.
- [14] Bui, B. D., Uchiyama, N., and Simba, K. R. (2016). Contouring control for three-axis machine tools based on nonlinear friction compensation for lead screws. *International Journal of Machine Tools and Manufacture*, 108:95–105.
- [15] Byrne, G., Dornfeld, D., and Denkena, B. (2003). Advancing cutting technology. *CIRP Annals*, 52(2):483–507.
- [16] Castro, H. and Burdekin, M. (2003). Dynamic calibration of the positioning accuracy of machine tools and coordinate measuring machines using a laser interferometer. *International Journal of Machine Tools and Manufacture*, 43(9):947–954.
- [17] Chen, H. and Sun, N. (2020). Nonlinear control of underactuated systems subject to both actuated and unactuated state constraints with experimental verification. *IEEE Transactions on Industrial Electronics*, 67(9):7702–7714.
- [18] Chen, H. and Sun, N. (2021). An output feedback approach for regulation of 5-dof offshore cranes with ship yaw and roll perturbations. *IEEE Transactions on Industrial Electronics*, pages 1–1.
- [19] Das, I. (1999). A preference ordering among various pareto optimal alternatives. *Structural Optimization*, 18(1):30–35.

- [20] Das, I. and Dennis, J. E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69.
- [21] Das, I. and Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657.
- [22] Deb, K. (2007). Current trends in evolutionary multi-objective optimization. *International Journal for Simulation and Multidisciplinary Design Optimization*, 1(1):1–8.
- [23] Deb, K. (2011). *Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction*, pages 3–34. Springer London, London.
- [24] Diaz, N., Helu, M., Jarvis, A., Tonissen, S., Dornfeld, D., and Schlosser, R. (2009). Strategies for minimum energy operation for precision machining. In *Machine Tool Technologies Research Foundation (MTTRF) 2009 Annual Meeting*, pages 47–50.
- [25] Dong, J., Ferreira, P., and Stori, J. (2007). Feed-rate optimization with jerk constraints for generating minimum-time trajectories. *International Journal of Machine Tools and Manufacture*, 47(12-13):1941–1955.
- [26] Duan, M. and Okwudire, C. E. (2016). Energy-efficient controller design for a redundantly actuated hybrid feed drive with application to machining. *IEEE/ASME Transactions on Mechatronics*, 21(4):1822–1834.
- [27] Erkorkmaz, K. and Altintas, Y. (2001). High speed CNC system design. part i: jerk limited trajectory generation and quintic spline interpolation. *International Journal of Machine Tools and Manufacture*, 41(9):1323–1345.
- [28] Erkorkmaz, K. and Altintas, Y. (2005). Quintic Spline Interpolation With Minimal Feed Fluctuation . *Journal of Manufacturing Science and Engineering*, 127(2):339–349.
- [29] Erkorkmaz, K. and Heng, M. (2008). A heuristic feedrate optimization strategy for nurbs toolpaths. *CIRP Annals*, 57(1):407–410.

- [30] Erkorkmaz, K., Yeung, C.-H., and Altintas, Y. (2006). Virtual cnc system. part ii. high speed contouring application. *International Journal of Machine Tools and Manufacture*, 46(10):1124–1138.
- [31] Fang, Y., Qi, J., Hu, J., Wang, W., and Peng, Y. (2020). An approach for jerk-continuous trajectory generation of robotic manipulators with kinematical constraints. *Mechanism and Machine Theory*, 153:103957.
- [32] Farina, M. and Amato, P. (2004). A fuzzy definition of "optimality" for many-criteria optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34(3):315–326.
- [33] Gill, P. E. and Wong, E. (2012). Sequential quadratic programming methods. In Lee, J. and Leyffer, S., editors, *Mixed Integer Nonlinear Programming*, pages 147–224, New York, NY. Springer New York.
- [34] Goldreich, O. (2008). Computational complexity: a conceptual perspective. *ACM Sigact News*, 39(3):35–39.
- [35] Halevi, Y., Carpanzano, E., Montalbano, G., and Koren, Y. (2011). Minimum energy control of redundant actuation machine tools. *CIRP Annals - Manufacturing Technology*, 60(1):433–436.
- [36] Hashem, I., Telen, D., Nimmegeers, P., Logist, F., and Impe, J. V. (2017). 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*, 106:544–558. ESCAPE-26.
- [37] He, K., Tang, R., and Jin, M. (2017). Pareto fronts of machining parameters for trade-off among energy consumption, cutting force and processing time. *International Journal of Production Economics*, 185:113–127.
- [38] Hendrawan, Y. M., Simba, K. R., and Uchiyama, N. (2018). Iterative learning based trajectory generation for machine tool feed drive systems. *Robotics and Computer-Integrated Manufacturing*, 51:230–237.

- [39] Hernandez-Vazquez, J.-M., Garitaonandia, I., Fernandes, M. H., Muñoa, J., and Lacalle, L. N. L. d. (2018). A consistent procedure using response surface methodology to identify stiffness properties of connections in machine tools. *Materials*, 11(7):1220.
- [40] Ho, P. M., Uchiyama, N., Sano, S., Honda, Y., Kato, A., and Yonezawa, T. (2014). Simple motion trajectory generation for energy saving of industrial machines. *SICE Journal of Control, Measurement, and System Integration*, 7(1):29–34.
- [41] Huang, J., Du, X., and Zhu, L.-M. (2018). Real-time local smoothing for five-axis linear toolpath considering smoothing error constraints. *International Journal of Machine Tools and Manufacture*, 124:67–79.
- [42] Huang, X., Zhao, F., Tao, T., and Mei, X. (2021). A newly developed corner smoothing methodology based on clothoid splines for high speed machine tools. *Robotics and Computer-Integrated Manufacturing*, 70:102106.
- [43] Huo, F. and Poo, A.-N. (2013). Precision contouring control of machine tools. *The International Journal of Advanced Manufacturing Technology*, 64(1-4):319–333.
- [44] Jeong, S. Y., Choi, Y. J., Park, P., and Choi, S. G. (2005). Jerk limited velocity profile generation for high speed industrial robot trajectories. *IFAC Proceedings Volumes*, 38(1):595–600.
- [45] Johansson, K. and Canudas-de-Wit, C. (2008). Revisiting the lugre friction model. *IEEE Control Systems Magazine*, 28(6):101–114.
- [46] Khodaygan, S. (2014). Manufacturing error compensation based on cutting tool location correction in machining processes. *International Journal of Computer Integrated Manufacturing*, 27(11):969–978.
- [47] Kim, I. Y. and de Weck, O. L. (2005). Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization*, 29(2):149–158.
- [48] Kucuk, S. (2017). Optimal trajectory generation algorithm for serial and parallel manipulators. *Robotics and Computer-Integrated Manufacturing*, 48:219–232.

- [49] Lee, Y.-H., Hsu, S.-C., Chi, T.-Y., Du, Y.-Y., Hu, J.-S., and Tsao, T.-C. (2021). Industrial robot accurate trajectory generation by nested loop iterative learning control. *Mechatronics*, 74:102487.
- [50] Li, B., Zhang, H., Ye, P., and Wang, J. (2020). Trajectory smoothing method using reinforcement learning for computer numerical control machine tools. *Robotics and Computer-Integrated Manufacturing*, 61:101847.
- [51] Li, F., Jiang, Y., Li, T., and Ehmann, K. F. (2018). Compensation of dynamic mechanical tracking errors in ball screw drives. *Mechatronics*, 55:27–37.
- [52] Logist, F., Houska, B., Diehl, M., and Van Impe, J. (2010). Fast pareto set generation for nonlinear optimal control problems with multiple objectives. *Structural and Multidisciplinary Optimization*, 42(4):591–603.
- [53] Logist, F. and Van Impe, J. (2012). Novel insights for multi-objective optimisation in engineering using normal boundary intersection and (enhanced) normalised normal constraint. *Structural and Multidisciplinary Optimization*, 45(3):417–431.
- [54] Mattson, C. A., Mullur, A. A., and Messac, A. (2004). Smart pareto filter: obtaining a minimal representation of multiobjective design space. *Engineering Optimization*, 36(6):721–740.
- [55] Messac, A., Ismail-Yahaya, A., and Mattson, C. A. (2003). The normalized normal constraint method for generating the pareto frontier. *Structural and Multidisciplinary Optimization*, 25(2):86–98.
- [56] Mohammadi, M., Musa, S. N., and Bahreininejad, A. (2014). Optimization of mixed integer nonlinear economic lot scheduling problem with multiple setups and shelf life using metaheuristic algorithms. *Advances in Engineering Software*, 78:41–51.
- [57] Mori, K., Uchiyama, N., Honzu, T., Sano, S., and Takagi, S. (2009). Time-optimal trajectory generation and contouring control for machine tool feed drive systems. In *2009 IEEE Conference on Emerging Technologies & Factory Automation*, pages 1–5. Institute of Electrical and Electronics Engineers (IEEE).

- [58] Mostaghel, N. and Davis, T. (1997). Representations of coulomb friction for dynamic analysis. *Earthquake Engineering & Structural Dynamics*, 26(5):541–548.
- [59] Msukwa, M. R., Nshama, E. W., and Uchiyama, N. (2020a). Adaptive sliding mode control with feedforward compensator for energy-efficient and high-speed precision motion of feed drive systems. *IEEE Access*, 8:43571–43581.
- [60] Msukwa, M. R., Nshama, E. W., and Uchiyama, N. (2020b). Contouring control based on reference adjustment and uncertainty compensator for precision motion of industrial feed drive systems. In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, pages 89–94.
- [61] Msukwa, M. R. and Uchiyama, N. (2019). Design and experimental verification of adaptive sliding mode control for precision motion and energy saving in feed drive systems. *IEEE Access*, 7:20178–20186.
- [62] Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.
- [63] Nshama, E. W., Msukwa, M. R., and Uchiyama, N. (2021). A trade-off between energy saving and cycle time reduction by pareto optimal corner smoothing in industrial feed drive systems. *IEEE Access*, 9:23579–23594.
- [64] Nshama, E. W. and Uchiyama, N. (2018). Time and energy optimal trajectory generation in feed drive systems using kinematic corner smoothing with interrupted acceleration. In *2018 26th Mediterranean Conference on Control and Automation (MED)*, pages 102–107.
- [65] Oda, M., Uchiyama, N., and Sano, S. (2015). Optimal trajectory generation for energy saving in machine tool contouring motion and experimental verification. In *2015 IEEE/SICE International Symposium on System Integration (SII)*, pages 121–126. Institute of Electrical and Electronics Engineers (IEEE).
- [66] Park, J. (1996). Motion profile planning of repetitive point-to-point control for maximum energy conversion efficiency under acceleration conditions. *Mechatronics*, 6(6):649–663.
- [67] Prévost, D., Lavernhe, S., Lartigue, C., and Dumur, D. (2011). Feed drive modelling for the simulation of tool path tracking in multi-axis high speed machining. *International Journal of Mechatronics and Manufacturing Systems*, 4(3-4):266–284.

- [68] Robert, K. W., Parris, T. M., and Leiserowitz, A. A. (2005). What is sustainable development? goals, indicators, values, and practice. *Environment: Science and Policy for Sustainable Development*, 47(3):8–21.
- [69] Sanchis, J., Martinez, M., Blasco, X., and Salcedo, J. (2008). A new perspective on multiobjective optimization by enhanced normalized normal constraint method. *Structural and Multidisciplinary Optimization*, 36(5):537–546.
- [70] Schittkowski, K. (2009). An active set strategy for solving optimization problems with up to 200,000,000 nonlinear constraints. *Applied Numerical Mathematics*, 59(12):2999–3007.
- [71] Sencer, B., Altintas, Y., and Croft, E. (2008). Feed optimization for five-axis cnc machine tools with drive constraints. *International Journal of Machine Tools and Manufacture*, 48(7):733–745.
- [72] Sencer, B., Ishizaki, K., and Shamoto, E. (2014). 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*, 76(9-12):1977–1992.
- [73] Sencer, B., Ishizaki, K., and Shamoto, E. (2015). High speed cornering strategy with confined contour error and vibration suppression for cnc machine tools. *CIRP Annals*, 64(1):369–372.
- [74] Sencer, B., Kakinuma, Y., and Yamada, Y. (2020). Linear interpolation of machining toolpaths with robust vibration avoidance and contouring error control. *Precision Engineering*, 66:269–281.
- [75] Sencer, B. and Tajima, S. (2017). Frequency optimal feed motion planning in computer numerical controlled machine tools for vibration avoidance. *Journal of Manufacturing Science and Engineering*, 139(1). 011006.
- [76] Shahzadeh, A., Khosravi, A., Robinette, T., and Nahavandi, S. (2018). Smooth path planning using biclothoid fillets for high speed cnc machines. *International Journal of Machine Tools and Manufacture*, 132:36–49.
- [77] Shi, J., Bi, Q., Zhu, L., and Wang, Y. (2015). Corner rounding of linear five-axis tool path by dual ph curves blending. *International Journal of Machine Tools and Manufacture*, 88:223–236.

- [78] Sihag, N. and Sangwan, K. S. (2020). A systematic literature review on machine tool energy consumption. *Journal of Cleaner Production*, 275:123125.
- [79] Simba, K. R., Heppeler, G., Bui, B. D., Hendrawan, Y. M., Sawodny, O., and Uchiyama, N. (2017). Bézier curve based trajectory generation and nonlinear friction compensation for feed drive contouring control. *IFAC-PapersOnLine*, 50(1):1944–1951. 20th IFAC World Congress.
- [80] Sindhya, K., Miettinen, K., and Deb, K. (2013). A hybrid framework for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(4):495–511.
- [81] Song, D. N., Ma, J. W., Zhong, Y. G., and Yao, J. J. (2021). Global smoothing of short line segment toolpaths by control-point-assigning-based geometric smoothing and fir filtering-based motion smoothing. *Mechanical Systems and Signal Processing*, 160:107908.
- [82] Sun, S. and Altintas, Y. (2021). A g3 continuous tool path smoothing method for 5-axis cnc machining. *CIRP Journal of Manufacturing Science and Technology*, 32:529–549.
- [83] Sun, Y.-H., Chen, T., Wu, C. Q., and Shafai, C. (2016). A comprehensive experimental setup for identification of friction model parameters. *Mechanism and Machine Theory*, 100:338–357.
- [84] Tajima, S. and Sencer, B. (2016). Kinematic corner smoothing for high speed machine tools. *International Journal of Machine Tools and Manufacture*, 108:27–43.
- [85] Tajima, S. and Sencer, B. (2017). Global tool-path smoothing for cnc machine tools with uninterrupted acceleration. *International Journal of Machine Tools and Manufacture*, 121:81–95. Special Issue on the State-of-the-Art in North American Manufacturing Research.
- [86] Tajima, S. and Sencer, B. (2020). Real-time trajectory generation for 5-axis machine tools with singularity avoidance. *CIRP Annals*, 69(1):349–352.
- [87] Teichert, K., Currie, G., Küfer, K.-H., Miguel-Chumacero, E., Süß, P., Walczak, M., and Currie, S. (2019). Targeted multi-criteria optimisation in imrt planning supplemented by knowledge based model creation. *Operations Research for Health Care*, 23:100185.

- [88] Teunter, R., Tang, O., and Kaparis, K. (2009). Heuristics for the economic lot scheduling problem with returns. *International Journal of Production Economics*, 118(1):323–330. Special Section on Problems and models of inventories selected papers of the fourteenth International symposium on inventories.
- [89] Tulsyan, S. and Altintas, Y. (2015). Local toolpath smoothing for five-axis machine tools. *International Journal of Machine Tools and Manufacture*, 96:15–26.
- [90] Uchiyama, N. (2008). Adaptive two-degree-of-freedom control of feed drive systems. *International Journal of Machine Tools and Manufacture*, 48(3):437–445.
- [91] Uchiyama, N., Goto, K., and Sano, S. (2015). Analysis of energy consumption in fundamental motion of industrial machines and experimental verification. In *2015 American Control Conference*, pages 2179–2184.
- [92] Uchiyama, N., Honda, Y., and Sano, S. (2014). Residual vibration suppression and energy saving in industrial machines using a trapezoidal velocity profile. In *2014 American Control Conference*, pages 323–328.
- [93] Uchiyama, N., Mori, K., Terashima, K., Saeki, T., Kamigaki, T., and Kawamura, H. (2013). Optimal motion trajectory generation and real-time trajectory modification for an industrial robot working in a rectangular space. *Journal of System Design and Dynamics*, 7(3):278–292.
- [94] Uchiyama, N. and Sano, S. (2010). Robust adaptive contouring control for biaxial feed drive systems. In *SPEEDAM 2010*, pages 937–941. Institute of Electrical and Electronics Engineers (IEEE).
- [95] Vardhan, T. V. and Babu, B. S. (2020). Chapter nine - multiaxis cnc programming and machining. In *Modern Manufacturing Processes*, Woodhead Publishing Reviews: Mechanical Engineering Series, pages 167–175. Woodhead Publishing.
- [96] Wan, M., Qin, X.-B., Xiao, Q.-B., Liu, Y., and Zhang, W.-H. (2021). Asymmetrical pythagorean-hodograph (ph) spline-based c3 continuous corner smoothing algorithm for five-axis tool paths with short segments. *Journal of Manufacturing Processes*, 64:1387–1411.

- [97] Wang, W., Hu, C., Zhou, K., and He, S. (2019). (b.6) corner trajectory smoothing with asymmetrical transition profile for cnc machine tools. *International Journal of Machine Tools and Manufacture*, 144:103423.
- [98] Wang, W., Hu, C., Zhou, K., He, S., and Zhu, L. (2021). Local asymmetrical corner trajectory smoothing with bidirectional planning and adjusting algorithm for cnc machining. *Robotics and Computer-Integrated Manufacturing*, 68:102058.
- [99] Wang, Z., Zhang, X., He, L., and Liu, T. (2020). Model predictive contouring control for biaxial feed drive system. In *2020 Chinese Control And Decision Conference (CCDC)*, pages 5237–5242.
- [100] Xiao, Q.-B., Wan, M., Liu, Y., Qin, X.-B., and Zhang, W.-H. (2020). Space corner smoothing of cnc machine tools through developing 3d general clothoid. *Robotics and Computer-Integrated Manufacturing*, 64:101949.
- [101] Xiao, Q. B., Wan, M., Qin, X. B., Liu, Y., and Zhang, W. H. (2021). Real-time smoothing of g01 commands for five-axis machining by constructing an entire spline with the bounded smoothing error. *Mechanism and Machine Theory*, 161:104307.
- [102] Xie, F., Chen, L., Li, Z., and Tang, K. (2020). Path smoothing and feed rate planning for robotic curved layer additive manufacturing. *Robotics and Computer-Integrated Manufacturing*, 65:101967.
- [103] Xie, Z., Xie, F., Liu, X. J., and Wang, J. (2021). Global g3 continuity toolpath smoothing for a 5-dof machining robot with parallel kinematics. *Robotics and Computer-Integrated Manufacturing*, 67:102018.
- [104] Yan, J. and Li, L. (2013). Multi-objective optimization of milling parameters – the trade-offs between energy, production rate and cutting quality. *Journal of Cleaner Production*, 52:462–471.
- [105] Yan, Y., Zhang, L., and Zhang, K. (2016). Corner smoothing transition algorithm for five-axis linear tool path. *Procedia CIRP*, 56:604–609.
- [106] Yang, J. and Yuen, A. (2017). An analytical local corner smoothing algorithm for five-axis cnc machining. *International Journal of Machine Tools and Manufacture*, 123:22–35.

- [107] Zhou, L., Li, J., Li, F., Meng, Q., Li, J., and Xu, X. (2016). Energy consumption model and energy efficiency of machine tools: a comprehensive literature review. *Journal of Cleaner Production*, 112:3721–3734.

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.

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

International journal papers

- Nshama, E. W., Msukwa, M.R., and Uchiyama, N. (2021). A trade-off between energy saving and cycle time reduction by Pareto optimal corner smoothing in industrial feed drive systems. *IEEE Access*, 9: 23579-23594.
- Nshama, E. W. and Uchiyama N. (2021). Pareto optimization of cycle time and motion accuracy in trajectory planning for industrial feed drive systems. *IEEE Access*, 9: 114104-114119.

International conference papers

- Nshama, E. W., Msukwa, M.R., and Uchiyama, N. (2019). Pareto optimization of energy and tolerance in motion trajectory generation for industrial feed drive systems. In *2019 American Control Conference (ACC)*, pages 842-847.
- Nshama, E. W. and Uchiyama N. (2018). Time and energy optimal trajectory generation in feed drive systems using kinematic corner smoothing with interrupted acceleration. In *2018 26th Mediterranean Conference on Control and Automation (MED)*, pages 102-107.