

Received July 7, 2021, accepted August 9, 2021, date of publication August 16, 2021, date of current version August 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104935

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

ENOCK WILLIAM NSHAMA<sup>1,2</sup> AND NAOKI UCHIYAMA<sup>1</sup>, (Member, IEEE)

<sup>1</sup>Department of Mechanical Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan

<sup>2</sup>Department of Mechanical Engineering, University of Dar es Salaam, Dar es Salaam 35131, Tanzania

Corresponding author: Enock William Nshama (enocknshama@gmail.com)

This work was supported in part by the Machine Tool Technologies Research Foundation, San Francisco, CA, USA, in part by Magnescale Company Ltd., Kanagawa, Japan, in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, and in part by Japan Society for the Promotion of Science (JSPS) KAKENHI (8桁の課題番号) under Grant JP20K04361.

**ABSTRACT** Manufacturing industries aim to improve product quantity and quality. These trade-off objectives are typically manifested as cycle time reduction and motion accuracy improvement. Corner smoothing approaches that reduce the cycle time of piecewise linear trajectories are proposed in the literature. This study tackles the two objectives by Pareto-optimal corner smoothing with constraints imposed as kinematic limits, continuity conditions and user-specified cornering tolerance. Linear and cornering motions along a contour are respectively described by jerk-limited acceleration profiles and a modified kinematic corner smoothing with interrupted acceleration (KCSIA) approach. 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 effectiveness of the proposed method is validated through experiments, where the best trade-off solution reduces the maximum and average contouring errors by 47.53 % and 25.40 % while it increases cycle time by 2.53 % compared to KCSIA.

**INDEX TERMS** Trajectory generation, motion accuracy, corner smoothing, Pareto optimization, industrial feed drives systems.

## I. INTRODUCTION

Manufacturing industries typically use computer numerical control (CNC) machine tools due to their accuracy, repeatability and speed in performing tasks [1]. Feed drive systems (FDSs) actuate CNC machine tools' motion axes [2]. 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 [3] and contouring [4], [5] errors in feed drive systems. Chen and Sun propose a nonlinear controller for underactuated systems [6]. A feedback control strategy is proposed in [7] 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.

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson<sup>1</sup>.

Trajectory generation methods have been proposed in the literature to reduce errors and cycle time. Feedforward compensation strategies such as iterative learning control [8], [9] and neural networks and reinforcement learning [10] 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 [11]. Sencer *et al.* propose time-optimal feed scheduling along B-spline tool paths for 5-axis CNC machine tools [12]. Frequency-optimal acceleration profiles are proposed by Sencer and Tajima for vibration suppression [13]. Kucuk proposes minimum time trajectory generation using cubic spline and 7<sup>th</sup> order polynomial interpolations [14]. 15-phase sinusoidal jerk profiles are proposed by Fang *et al.* for cycle time reduction and high-frequency harmonic suppression [15]. Wang *et al.* propose time-optimal S-curve velocity profile generation for robotic arms [16]. In economic lot scheduling of supply chains, a sub-optimal cycle time of a process lot can be selected to reduce overall production

costs [17], [18]. To this end, Jeong *et al.* present time-optimal and time-fixed jerk-limited velocity profile generation algorithms [19]. Besset and Béarée propose online finite impulse response-based trajectory generation for time-optimal, fixed time and jerk-time fixed cases [20].

Before a contour is fed to feed drive systems, 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 a pair of adjacent segments [21]. Global corner smoothing is normally used for motion planning along short-segment paths [22]–[24]. However, it is more challenging to evaluate and constrain the smoothing error for global corner smoothing compared to local corner smoothing [25].

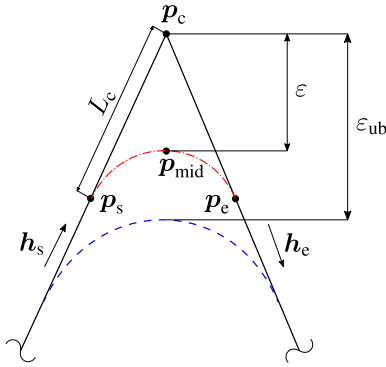
Local corner smoothing can be categorized as geometric and kinematic local corner smoothing, where geometric local corner smoothing separately considers geometric and kinematic constraints in two steps while kinematic local corner smoothing plans smooth velocity transitions directly by considering both constraints in one step [26]–[28]. In literature, geometric local corner smoothing using B-spline [29]–[31], Pythagorean Hodograph [32] and Bézier [27] curves are proposed. Regarding kinematic local corner smoothing, kinematic corner smoothing with interrupted acceleration (KCSIA) has been proposed for cycle time optimality [33]. An energy-time trade-off using KCSIA is studied in [34]. Finite impulse response-based kinematic local corner smoothing has been proposed for reducing time and vibrations [35], [36]. Kinematic local corner smoothing approaches using clothoids [37]–[39] and asymmetrical double constant-jerk cornering profiles [21] have been proposed for improving cycle time. Regarding CNC machining, there are several different cases of requirements for surface quality and cycle time: a high-accuracy case, a 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 [27], [40]. Under sub-optimal cycle time scenarios, driving these errors to their upper limits is no longer necessary. Thus, various trade-off solutions for cycle time and cornering error can be selected by a decision-maker.

A Pareto set of a multi-objective optimization problem (MOOP) can be generated by vectorization or scalarization methods [41], [42]. 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 single-objective

optimization problems (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 solutions in non-convex Pareto regions [43]. Approaches such as normal boundary intersection [44], normalized normal constraint [45], [46] methods have been proposed. The normal boundary intersection approach and, to a lesser degree, the normalized normal constraint are prone to generating non-Pareto optimal solutions. Logist and Van Impe propose a criterion for detecting such solutions [47]. The scalarization methods typically distribute the SOOPs evenly, resulting in a waste of computational effort 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 [48]. 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 [42].

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 normalized normal constraint method is used to formulate the bi-objective optimization problem 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 jerk-limited acceleration profiles (JLAP) [49] 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 [42] 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 normalized normal constraint method. Each Pareto-optimal solution is computed using sequential quadratic programming [50]. 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.



**FIGURE 1.** An illustration of a cornering path generated by a corner smoothing method. A corner point  $p_c$  is smoothed by a red dash-dot curve with a cornering error  $\varepsilon$  bounded by  $0 < \varepsilon \leq \varepsilon_{ub}$ .

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 paper is organized as follows: Section II illustrates the design of Pareto-optimal trajectories, Section III presents the optimization and experimental results. A discussion on the findings and concluding remarks are described in Sections IV and V, respectively.

## II. 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.

### A. PROBLEM FORMULATION

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 [22], [27], [40]. Hence, motion accuracy improvement and cycle time reduction are conflicting objectives in trajectory generation for piecewise linear contours.

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. 1, where  $p_c$  is the original corner point and the cornering error  $\varepsilon$  is restricted by upper bound  $\varepsilon_{ub}$ . The cornering path from position vector  $p_s$  to  $p_e$  is symmetrical about line  $\overline{p p_{mid}}$ , where the distance between the start/end of the path and the original corner  $p_c$  is the cornering Euclidean distance  $L_c$ .  $h_s$  and  $h_e$  are the direction vectors at the start and end of the path, respectively. The minimization of  $\varepsilon$  and reduction of  $L_c$  are non-conflicting objectives. Hence, a SOOP

$$\min_{\mu_{c,m}} L_{c,m}(\mu_{c,m}) \quad (1)$$

$$\begin{aligned} \text{subject to } & \mathbf{g}(\mu_{c,m}) = \mathbf{0}, \\ & \mathbf{q}(\mu_{c,m}) \leq \mathbf{0}, \\ & -\mu_{c,m} \leq \mathbf{0} \end{aligned} \quad (2)$$

is used to describe the smoothing minimization objective for the  $m^{\text{th}}$  corner, where  $\mu_{c,m}$  is parameter vector that consists of variables that describe a cornering trajectory. As detailed in Section II-C2, 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

$$T_{\text{cycle}} = \sum_{m=1}^{n_l} T_{l,m}(\mu_{l,m}) + \sum_{m=1}^{n_c} T_{c,m}(\mu_{c,m}) \quad (3)$$

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.  $\mu_{l,m}$  is a parameter vector that describes the  $m^{\text{th}}$  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

$$\begin{aligned} \min_{\mu} \{ & L_{c,\text{tot}}(\mu), T_{\text{cycle}}(\mu) \} \\ \mu = & [\mu_{l,1}, \mu_{c,1}, \mu_{l,2}, \mu_{c,2}, \dots, \mu_{l,n_l}, \mu_{c,n_c}]^T \quad (4) \\ \text{subject to } & \mathbf{g}(\mu) = \mathbf{0}, \\ & \mathbf{q}(\mu) \leq \mathbf{0}, \\ & -\mu \leq \mathbf{0} \end{aligned} \quad (5)$$

with

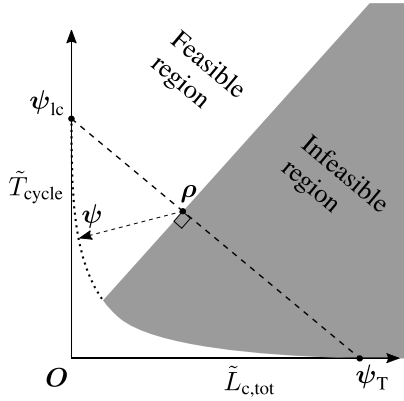
$$L_{c,\text{tot}}(\mu) = \sum_{m=1}^{n_c} L_{c,m}(\mu_{c,m}), \quad (6)$$

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.  $\mu$  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 (4) as:

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

$$\left. \begin{aligned} L_{c,\text{max}} &= L_{c,\text{tot}}(\mu_T) \\ T_{\text{min}} &= T_{\text{cycle}}(\mu_T) \end{aligned} \right\}, \mu_T = \arg \min_{\mu} T_{\text{cycle}}(\mu). \quad (7)$$



**FIGURE 2.** An illustration of the reformulated optimization problem (9) with the feasible region reduced by the normalized normal constraint inequality constraint (10). The utopia point is marked by the origin  $O$ .

In a normalized objective space,

$$\begin{aligned}\tilde{L}_{c,tot}(\boldsymbol{\mu}) &= \frac{L_{c,tot}(\boldsymbol{\mu}) - L_{c,min}}{\Delta L_{c,tot}}, \\ \tilde{T}_{cycle}(\boldsymbol{\mu}) &= \frac{T_{cycle}(\boldsymbol{\mu}) - T_{min}}{\Delta T_{cycle}}\end{aligned}\quad (8)$$

describe the respective individual objectives in (4), where  $\Delta L_{c,tot} = L_{c,max} - L_{c,min}$  and  $\Delta T_{cycle} = T_{max} - T_{min}$  are the saving potentials in corner smoothing and cycle time, respectively. In accordance with the normalized normal constraint method [45], (4) is reformulated as:

$$\min_{\boldsymbol{\mu}} \tilde{T}_{cycle}(\boldsymbol{\mu}) \quad (9)$$

subject to (5) and an additional normalized normal constraint inequality constraint

$$[\boldsymbol{\psi}_T - \boldsymbol{\psi}_{1c}]^T [\boldsymbol{\psi} - \boldsymbol{\rho}] \leq 0 \quad (10)$$

that limits the feasible region in the normalized objective space (Fig. 2) with

$$\begin{aligned}\boldsymbol{\psi} &= [\tilde{L}_{c,tot}(\boldsymbol{\mu}), \tilde{T}_{cycle}(\boldsymbol{\mu})]^T, \\ \boldsymbol{\psi}_{1c} &= [\tilde{L}_{c,tot}(\boldsymbol{\mu}_{1c}), \tilde{T}_{cycle}(\boldsymbol{\mu}_{1c})]^T, \\ \boldsymbol{\psi}_T &= [\tilde{L}_{c,tot}(\boldsymbol{\mu}_T), \tilde{T}_{cycle}(\boldsymbol{\mu}_T)]^T, \\ \boldsymbol{\rho} &= (1 - \zeta) \boldsymbol{\psi}_{1c} + \zeta \boldsymbol{\psi}_T, \\ 0 &\leq \zeta \leq 1,\end{aligned}\quad (11)$$

where a point  $\boldsymbol{\psi}$  on a Pareto frontier  $\Psi$  corresponds to the point  $\boldsymbol{\rho}$  on the convex hull of individual minima obtained at a weighting factor  $\zeta$ .

$\Psi$  is generated using the divide and conquer recursive structure [42], where (9) is solved by sequence quadratic programming [50] for different  $\zeta$ . In this work, the Pareto optimal point that is nearest to the utopia point is considered to be the best trade-off solution

$$\boldsymbol{\psi}^* = \arg \min_{\boldsymbol{\psi}} \|\boldsymbol{\Psi}\|_2 \quad (12)$$

obtained at the weighting factor  $\zeta^*$  which corresponds to the parameter vector  $\boldsymbol{\mu}^*$  in the decision space. The optimization process described above is demonstrated as Algorithm 1 in the appendix, where it receives a piecewise linear tool path a series of 2D points  $\boldsymbol{p}_c$ s, cornering tolerance  $0 < \varepsilon \leq \varepsilon_{ub}$  and kinematic limits for each axis.

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

$$\min_{\boldsymbol{\mu}} L_{c,tot}(\boldsymbol{\mu}) \quad (13)$$

subject to (5) and a cycle time constraint

$$T_{cycle}(\boldsymbol{\mu}) - T_{fixed}(\boldsymbol{\mu}) = 0 \quad (14)$$

where  $T_{fixed}$  is the fixed cycle time defined by a user.

## B. OPTIMIZATION CONSTRAINTS

In order to ensure that a Pareto optimal solution is implementable in a real FDS, kinematic limitations for the  $k^{\text{th}}$  axis must be incorporated as optimization constraints. Hence, kinematic constraints

$$\left. \begin{aligned}(\ddot{r}_k(t))^2 - j_{lim,k}^2 &\leq 0 \\ (\ddot{r}_k(t))^2 - a_{lim,k}^2 &\leq 0 \\ (\dot{r}_k(t))^2 - v_{lim,k}^2 &\leq 0\end{aligned}\right\}, \quad \forall t \quad (15)$$

are defined, where  $r_k$  is a trajectory position and  $j_{lim,k}$ ,  $a_{lim,k}$ , and  $v_{lim,k}$  are the respective jerk, acceleration and velocity limits. (15) is described in a quadratic form so as to guarantee differentiability  $\forall \boldsymbol{\mu} \geq 0$ . At each corner, the  $m^{\text{th}}$  cornering error is bounded by the constraints

$$\varepsilon_m - \varepsilon_{ub,m} \leq 0. \quad (16)$$

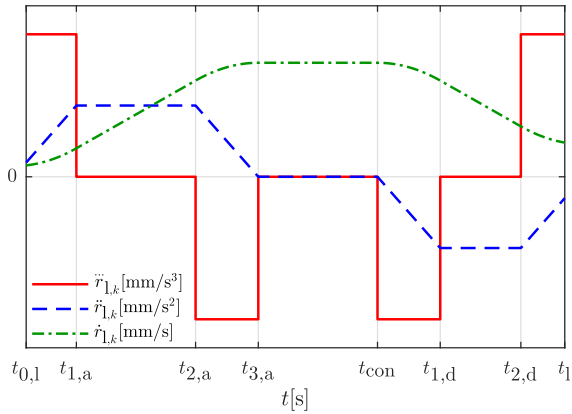
Geometric constraints

$$\|\boldsymbol{p}_{c,m} - \boldsymbol{p}_{c,m-1}\|_2 - [L_{c,m-1} + s_m + L_{c,m}] = 0 \quad (17)$$

are defined in order to avoid overlapping the  $(m-1)^{\text{th}}$  and  $m^{\text{th}}$  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

$$\begin{aligned}\boldsymbol{r}_1^{(n)}(t_{ms}) - \boldsymbol{r}_c^{(n)}(t_{ms}) &= \mathbf{0}, \\ \boldsymbol{r}_1^{(n-1)}(t_{ms}) - \boldsymbol{r}_c^{(n-1)}(t_{ms}) &= \mathbf{0}, \\ &\vdots \\ \dot{\boldsymbol{r}}_1(t_{ms}) - \dot{\boldsymbol{r}}_c(t_{ms}) &= \mathbf{0}, \\ \boldsymbol{r}_1(t_{ms}) - \boldsymbol{r}_c(t_{ms}) &= \mathbf{0}\end{aligned}\quad (18)$$

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



**FIGURE 3.** A  $k^{\text{th}}$  axis jerk limited acceleration profile demonstration. Jerk, acceleration and velocity are accordingly labeled as  $\ddot{r}_{1,k}$ ,  $\dot{r}_{1,k}$  and  $r_{1,k}$ .

### C. 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.

#### 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 [33], [49]. Fig. 3 shows the JLAP motion phases. A JLAP jerk profile

$$\ddot{r}_{1,k}(t) = \begin{cases} j_{\max,k}, & t_{0,1} \leq t < t_{1,a}, \\ 0, & t_{1,a} \leq t < t_{2,a}, \\ -j_{\max,k}, & t_{2,a} \leq t < t_{3,a}, \\ 0, & t_{3,a} \leq t < t_{\text{con}}, \\ -j_{\max,k}, & t_{\text{con}} \leq t < t_{1,d}, \\ 0, & t_{1,d} \leq t < t_{2,d}, \\ j_{\max,k}, & t_{2,d} \leq t < t_1 \end{cases} \quad (19)$$

is defined, where  $j_{\max,k}$  is the  $k^{\text{th}}$  axis maximum jerk magnitude in the motion. The time intervals  $t_{0,1} \leq t < t_{3,a}$ ,  $t_{3,a} \leq t < t_{\text{con}}$  and  $t_{\text{con}} \leq t < t_1$  are respectively the acceleration, constant velocity and deceleration phases. The total motion duration

$$T_1 = T_{1,a} + T_{2,a} + T_{3,a} + T_{\text{con}} + T_{1,d} + T_{2,d} + T_{3,d} \quad (20)$$

is obtained as a sum of time intervals

$$\begin{aligned} T_{1,a} &= t_{1,a} - t_{0,1}, & T_{1,d} &= t_{1,d} - t_{\text{con}}, \\ T_{2,a} &= t_{2,a} - t_{1,a}, & T_{2,d} &= t_{2,d} - t_{1,d}, \\ T_{3,a} &= t_{3,a} - t_{2,a}, & T_{3,d} &= t_1 - t_{2,d} \\ T_{\text{con}} &= t_{\text{con}} - t_{3,a}, \end{aligned} \quad (21)$$

that are computed according to kinematic constraints (15) and boundary conditions (18). By successive integration of (19),

the  $k^{\text{th}}$  axis total displacement

$$\begin{aligned} \Delta r_{1,k} &= v_{s,1,k} T_1 + \frac{1}{2} a_{s,1,k} T_1^2 + j_{\max,k} \\ &\times \left[ \frac{1}{6} (T_{1,a} - T_{3,a})^3 \right. \\ &+ \frac{1}{2} T_{1,a} (T_{1,a} + T_{2,a}) (T_{2,a} + 2T_{3,a}) \\ &- \frac{1}{6} (T_{1,d} - T_{3,d})^3 \\ &- \frac{1}{2} T_{1,d} (T_{1,d} + T_{2,d}) (T_{2,d} + 2T_{3,d}) \\ &+ (T_{\text{con}} + T_d) \left\{ T_{1,a} (T_{2,a} + T_{3,a}) \right. \\ &\left. \left. + \frac{1}{2} (T_{1,a} - T_{3,a}) (T_{1,a} + T_{3,a} + T_{\text{con}} + T_d) \right\} \right] \quad (22) \end{aligned}$$

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

$$\boldsymbol{\mu}_1 = [T_{1,a}, T_{2,a}, T_{3,a}, T_{\text{con}}, T_{1,d}, T_{2,d}, T_{3,d}]. \quad (23)$$

For JLAPs, kinematic constraints (15) are implemented as:

$$\left. \begin{aligned} j_{\max,m,k}^2 - j_{\lim,k}^2 &= 0 \\ (a_{s,1,k} + j_{\max,m,k} T_{1,a,m})^2 - a_{\lim,k}^2 &\leq 0 \\ (j_{\max,m,k} T_{1,d,m})^2 - a_{\lim,k}^2 &\leq 0 \\ v_{\text{con},m,k}^2 - v_{\lim,k}^2 &\leq 0 \end{aligned} \right\}, \quad m \in \{1, 2, \dots, n_1\}, \quad (24)$$

with a maximum  $k^{\text{th}}$  axis velocity

$$\begin{aligned} v_{\text{con},m,k} &= v_{s,1,m,k} + a_{s,1,m,k} (T_{1,a,m} + T_{2,a,m} + T_{3,a,m}) \\ &+ \frac{1}{2} j_{\max,m,k} (T_{1,a,m}^2 - T_{3,a,m}^2) \\ &+ j_{\max,m,k} T_{1,a,m} (T_{2,a,m} + T_{3,a,m}) \end{aligned} \quad (25)$$

along the  $m^{\text{th}}$  linear segment.

#### 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 [33]. Fig. 4 shows a KCSIA demonstration, where, based on JLAPs, its jerk profile

$$\ddot{r}_{c,k}(t) = \begin{cases} j_{c,k}, & t_{0,c} \leq t < t_{1,c}, \\ 0, & t_{1,c} \leq t < t_{2,c}, \\ -j_{c,k}, & t_{2,c} \leq t < t_c, \end{cases} \quad (26)$$

is defined with

$$\begin{aligned} j_{c,k} &= \frac{v_{e,c,k} - v_{s,c,k}}{T_{1,c}(T_{1,c} + T_{2,c})}, & T_{1,c} &= t_{1,c} - t_{0,c} \\ v_{s,c,k} &= V_c h_{s,k}, & &= t_c - t_{2,c}, \\ v_{e,c,k} &= V_c h_{e,k}, & T_{2,c} &= t_{2,c} - t_{1,c}. \\ a_{c,k} &= j_{c,k} T_{1,c}, \end{aligned} \quad (27)$$

$j_{c,k}$  is the cornering jerk. By successive integration of (26), a cornering path

$$\begin{aligned} r_{c,k}(t) &= \begin{cases} p_{s,k} + v_{s,c,k}(t - t_{0,c}) + \frac{1}{6}j_{c,k}(t - t_{0,c})^3, & t_{0,c} \leq t < t_{1,c}, \\ p_{1,k} + v_{s,c,k}(t - t_{1,c}) \\ + \frac{1}{2}j_{c,k}T_{1,c}(t - t_{1,c})(T_{1,c} + t - t_{1,c}), & t_{1,c} \leq t < t_{2,c}, \\ p_{2,k} + v_{s,c,k}(t - t_{2,c}) \\ + \frac{1}{2}j_{c,k}T_{1,c}(t - t_{2,c})(T_{1,c} + 2T_{2,c} + t - t_{2,c}) \\ - \frac{1}{6}j_{c,k}(t - t_{2,c})^3 & t_{2,c} \leq t < t_c \end{cases} \end{aligned} \quad (28)$$

is defined with

$$\begin{aligned} p_{1,k} &= p_{s,k} + v_{s,c,k}T_{1,c} + \frac{1}{6}j_{c,k}T_{1,c}^3, \\ p_{2,k} &= p_{1,k} + v_{s,c,k}T_{2,c} + \frac{1}{2}j_{c,k}T_{1,c}T_{2,c}(T_{1,c} + T_{2,c}), \end{aligned} \quad (29)$$

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

$$\Delta r_{c,k} = \left\{ v_{s,c,k} + \frac{1}{2}j_{c,k}T_{1,c}(T_{1,c} + T_{2,c}) \right\} T_c \quad (30)$$

is obtained from (28), where  $T_c = 2T_{1,c} + T_{2,c}$  is the total motion duration. The cornering Euclidean length

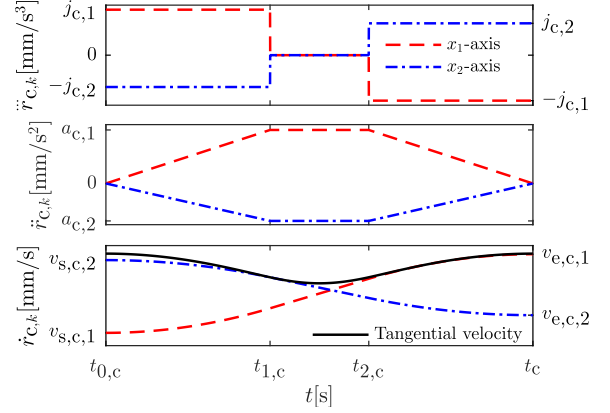
$$L_c = \frac{\Delta r_{c,k}}{(h_{s,k} + h_{e,k})} = \frac{1}{2}V_c T_c \quad (31)$$

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

$$\begin{aligned} p_{s,k} &= p_{c,k} - L_c h_{s,k}, \\ p_{\text{mid},k} &= p_{s,k} + v_{s,c,k} \left[ T_{1,c} + \frac{T_{2,c}}{2} \right] + \frac{1}{2}j_{c,k}T_{1,c} \left( \frac{T_{2,c}^2}{2} \right) \\ &+ j_{c,k} \left[ \frac{1}{6}T_{1,c}^3 + \frac{1}{2}T_{1,c}^2 \left( \frac{T_{2,c}}{2} \right) \right] \quad \text{and} \\ p_{e,k} &= p_{c,k} + L_c h_{e,k}, \end{aligned} \quad (32)$$

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

$$\varepsilon = \frac{V_c \left\{ T_{1,c}^2 + 3(T_{1,c} + T_{2,c})^2 \right\}}{24(T_{1,c} + T_{2,c})} \|\mathbf{h}_e - \mathbf{h}_s\|_2. \quad (33)$$



**FIGURE 4.** A demonstration of kinematic corner smoothing with interrupted acceleration, where  $k^{\text{th}}$  axis jerk, acceleration and velocity labeled as  $\ddot{r}_{c,k}$ ,  $\ddot{r}_{c,k}$  and  $\dot{r}_{c,k}$ , respectively.

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

$$\begin{aligned} \max_{\mu_c} & V_{c,m} \\ \mu_{c,m} &= [V_{c,m}, T_{1,c,m}, T_{2,c,m}] \end{aligned} \quad (34)$$

subject to (15)–(17) and  $-\mathbf{u}_{c,m} \leq 0$ . This is followed by the generation of the  $m^{\text{th}}$  cornering path, where the start/end points are obtained from (31) - (32) and the path is plotted using (28). Afterwards, motions along the line segments that connect the cornering paths are stitched using time optimal JLAPs.

In order to solve (9) or (13), (33) is incorporated in the cornering constraints (16). The kinematic constraints (15) are realized in the form of

$$\left. \begin{aligned} j_{c,m,k}^2 - j_{\text{lim},k}^2 &\leq 0 \\ a_{c,m,k}^2 - a_{\text{lim},k}^2 &\leq 0 \\ v_{s,c,m,k}^2 - v_{\text{lim},k}^2 &\leq 0 \\ v_{e,c,m,k}^2 - v_{\text{lim},k}^2 &\leq 0 \end{aligned} \right\}, \quad m \in \{1, 2, \dots, n_c\}. \quad (35)$$

The geometric constraints (17) are implemented in a decoupled form

$$\begin{aligned} p_{c,m,k} - p_{c,m-1,k} - [L_{c,m-1} + L_{c,m}] h_{s,m,k} \\ - \Delta r_{1,m,k} &= 0, \\ m &\in \{1, 2, \dots, n_1\}. \end{aligned} \quad (36)$$

$C^2$  continuity constraints are established by setting

$$\left. \begin{aligned} \mathbf{a}_{e,1,m} = \ddot{\mathbf{r}}_{c,m}(t_{0,c}) &= \mathbf{0} \\ \mathbf{a}_{s,1,m+1} = \ddot{\mathbf{r}}_{c,m}(t_c) &= \mathbf{0} \\ \mathbf{v}_{e,1,m} - \mathbf{v}_{s,c,m} &= \mathbf{0} \\ \mathbf{v}_{s,1,m+1} - \mathbf{v}_{e,c,m} &= \mathbf{0} \\ \mathbf{p}_{e,1,m} - \mathbf{p}_{s,m} &= \mathbf{0} \\ \mathbf{p}_{s,1,m+1} - \mathbf{p}_{e,m} &= \mathbf{0} \end{aligned} \right\}, \quad m \in \{1, 2, \dots, n_c\} \quad (37)$$

where  $\mathbf{a}_{e,1,m}$  and  $\mathbf{v}_{e,1,m}$  are the acceleration and velocity vectors at the end of the  $m^{\text{th}}$  linear motion.  $\mathbf{p}_{s,1,m}$  and  $\mathbf{p}_{e,1,m}$  are the start and end position vectors of the  $m^{\text{th}}$  linear motion. All

TABLE 1. Identified plant parameters.

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

solutions of (9) or (13) 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 (19) and (28), where the linear segment and cornering path variables are retrieved from  $\mu$ .

### 3) SPACE COMPLEXITY ANALYSIS

Space complexity is described as the memory resource required to execute a computation [51]. For an optimization problem with  $n_{\text{var}}$  variables and  $n_{\text{con}}$  constraints, a typical sequence quadratic programming solver requires a  $\mathcal{O}(n_{\text{var}}^2 + n_{\text{var}}n_{\text{con}})$  long double precision working array [52]. KCSIA\* consists of

$$n_{\text{var}} = n_1 \text{dim}(\mu_1) + n_c \text{dim}(\mu_c), \quad (38)$$

variables, where  $\text{dim}(\cdot)$  denotes vector dimension. According to (23) and (34),  $n_{\text{var}} = 10n_1 - 3$ . With reference to (16), (24) and (35)–(37),  $n_{\text{con}} = 32n_1 - 21$ . Hence, the KCSIA\* memory demand is a double precision working array of length  $\mathcal{O}(420n_1^2 - 366n_1 + 72) \approx \mathcal{O}(n_1^2)$ . In contrast, KCSIA has a  $\mathcal{O}(1)$  memory demand since it separately optimizes each corner and linear segment.

## III. RESULTS

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.

### A. OPTIMIZATION CONDITIONS AND EXPERIMENTAL SETUP

The industrial biaxial table in Fig. 5 with kinematic limits:  $J_{\text{lim}} = [50000, 50000] \text{ mm/s}^3$ ,  $a_{\text{lim}} = [1000, 1000] \text{ mm/s}^2$  and  $v_{\text{lim}} = [80, 80] \text{ mm/s}$  is used for verifying the proposed method. The cornering tolerance at each corner is specified as  $0 < \varepsilon \leq 200 \mu\text{m}$ . A sequence quadratic programming implementation in MATLAB® environment is used to solve (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.

The experimental setup in Fig. 5 is used to validate the optimization results. Motion along each axis is actuated by computer controlled 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

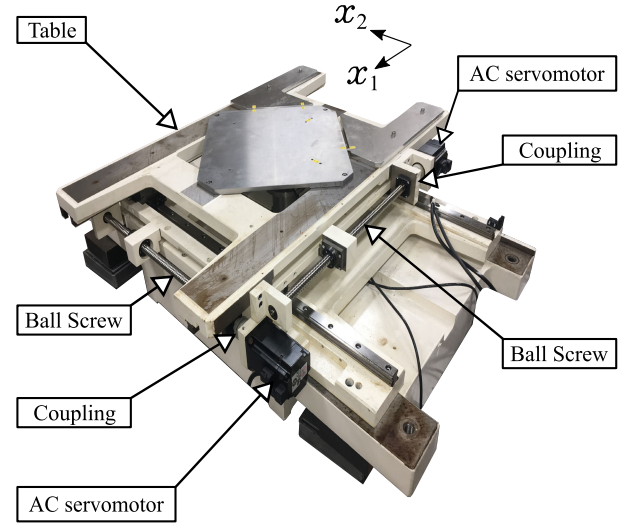


FIGURE 5. The industrial biaxial table used for experimental verification.

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. Reference trajectories are fed into a desktop computer, having Intel (R) Core i7-3770K CPU, 3.50 GHz, 8 GB 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.

In this study, the system dynamics of the industrial biaxial table are considered as a second-order model [53]

$$M\ddot{x}(t) + D\dot{x}(t) + F\text{sgn}\{\dot{x}(t)\} = u(t) \quad (39)$$

with

$$M = \text{diag}\{m_k\}, \quad D = \text{diag}\{d_k\}, \quad F = \text{diag}\{f_k\}, \\ K_F = \text{diag}\{k_{F,k}\}, \quad k \in \{1, 2\},$$

where  $M$ ,  $D$  and  $F$  are diagonal matrices of axial inertia term, viscous friction and Coulomb friction, respectively.  $u \in \mathbb{R}^2$  and  $x \in \mathbb{R}^2$  are the control input and axial position vectors, respectively.  $\text{sgn}\{\dot{x}\}$  is a function that returns a vector with the respective signs of the  $\dot{x}$  elements. In a similar manner to [54], the tracking error dynamics

$$e(t) = r(t) - x(t), \\ \ddot{e}(t) = \ddot{r}(t) + M^{-1}\{u(t) - D\dot{x}(t) - F\text{sgn}\{\dot{x}(t)\}\} \quad (40)$$

are described, where a PD tracking controller with Coulomb and viscous friction compensation

$$u(t) = \hat{M}\{\ddot{r}(t) + K_P e(t) + K_D \dot{e}(t)\} + \hat{D}\dot{x}(t) \\ + \hat{F}\text{sgn}\{\dot{x}(t)\} \quad (41)$$

is implemented.  $\hat{M} = \text{diag}\{\hat{m}_k\}$ ,  $\hat{D} = \text{diag}\{\hat{d}_k\}$  and  $\hat{F} = \text{diag}\{\hat{f}_k\}$  correspond to the identified nominal values of  $M$ ,  $D$  and  $F$ , respectively.  $K_P = \text{diag}\{k_{P,k}\}$  and  $K_D = \text{diag}\{k_{D,k}\}$

**TABLE 2. 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	
	$\zeta$	$L_{c,tot}$ mm	$T_{cycle}$ s	$L_{c,tot}$ %	$T_{cycle}$ %
KCSIA	-	7.202	3.513	-	-
KCSIA* : $T_{cycle}$ optimal	1	5.422	3.496	24.715	0.484
KCSIA* : $L_{c,tot}$ optimal	0	$4.614 \times 10^{-6}$	3.739	$\approx 100$	-6.433
KCSIA* : Best trade-off	$\zeta^* = 0.53125$	2.713	3.602	62.330	-2.533

**TABLE 3. Influence the minimum trade-off level on the optimization process.**

Min. trade-off level	Best trade-off solution $\psi^*$			Computation time
	Weighting factor	Total cornering Euclidean length	Cycle Time	
$\delta_{min}$	$\zeta^*$	$L_{c,tot}$ 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

**TABLE 4. Experimental results summary of contouring performance for the star-shaped tool path.**

Trajectory	Weighting factor	Contouring error			
		Maximum	Average	Reduction relative to KCSIA	
				Maximum %	Average %
$\zeta$	$\mu\text{m}$	$\mu\text{m}$			
KCSIA	-	198.48	14.99	-	-
KCSIA* : $T_{cycle}$ optimal	1	197.27	13.86	0.61	7.55
KCSIA* : $L_{c,tot}$ optimal	0	55.26	9.89	72.16	34.03
KCSIA* : Best trade-off	$\zeta^* = 0.53125$	104.15	11.18	47.53	25.40

are the proportional and derivative gain diagonal matrices. The controller (41) is used as the control law for conducting experiments with identified plant parameters (see Table 1),  $k_{P,k} = 7225 \text{ s}^{-2}$  and  $k_{D,k} = 170 \text{ s}^{-1}$  for  $\forall k$ .

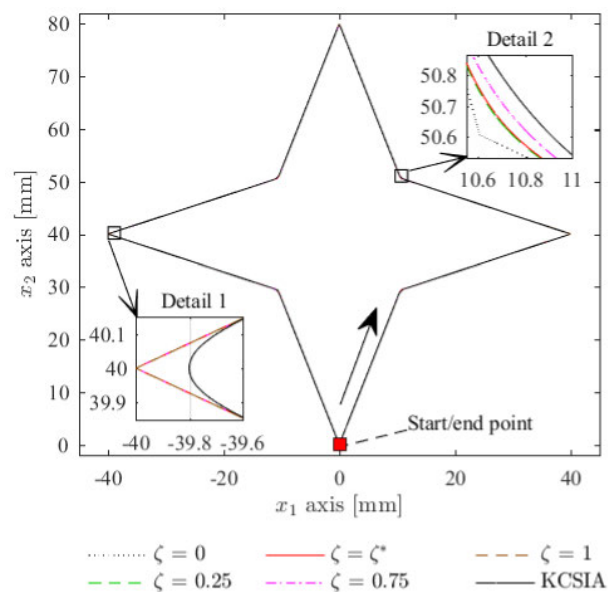
In order to verify the effectiveness of the proposed method, an experiment is conducted on the industrial biaxial table. KCSIA and  $\zeta = \{0, 0.25, \zeta^*, 0.75, 1\}$  KCSIA\* motion profiles are fed into the experimental setup as reference trajectories, where 10 trials are conducted to check for the repeatability of results for each trajectory. In this work, the contouring error estimation method in [40] is used, where contouring errors are computed with respect to the original piecewise linear tool paths. In this study, star-shaped (Section III-B) and complex (Section III-C) tool paths are considered for optimal trajectory generation and experimental verification.

### B. CASE I: STAR-SHAPED TOOL PATH

The star-shaped tool path in Fig. 6 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.

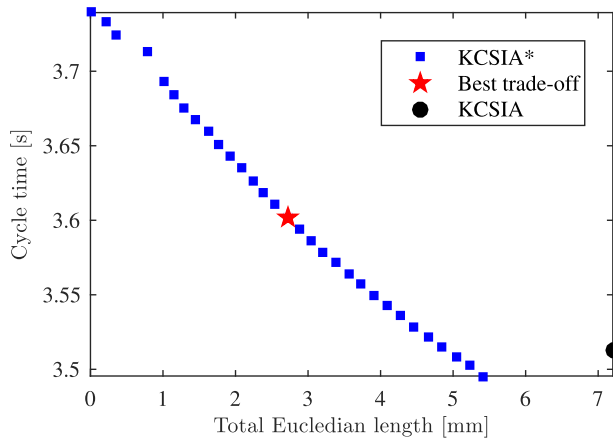
#### 1) OPTIMIZATION RESULTS

The KCSIA\* Pareto frontier is shown in Fig. 7, where KCSIA is a dominated solution. The time-optimal KCSIA\* solution

**FIGURE 6. The star-shaped path 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  $\zeta$  for the acute and obtuse corners, respectively.**

is a 0.017 s (i.e., 0.484 %) faster cycle time and 1.780 mm (i.e., 24.715 %) less corner smoothing than KCSIA. KCSIA





**FIGURE 7. Optimization results of the star-shaped tool path represented as a Pareto frontier.**

has an inferior solution since it maximizes cornering velocities, consequently maximizing the cornering Euclidean lengths as shown in (31) and (34). 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.

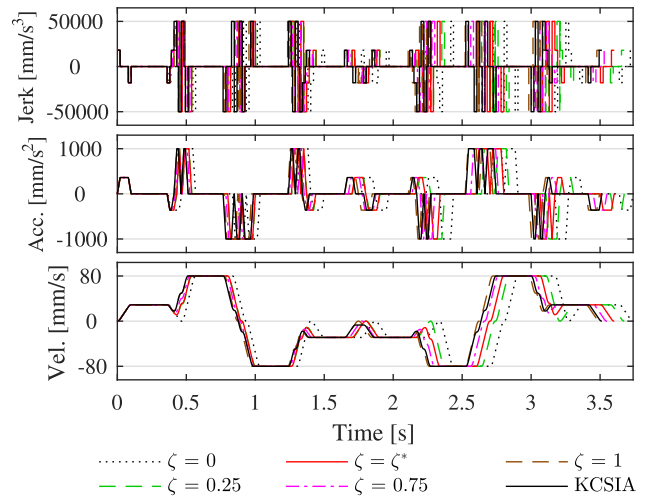
The best trade-off KCSIA\* solution is located at  $\zeta^* = 0.53125$ . It offers 2.709 mm less corner smoothing than the KCSIA\* time-optimal result while being 0.137 s faster than the minimum corner smoothing result. This corresponds to achieving 53.086 % and 53.006 % of the available cycle time and corner smoothing saving potentials. The optimization results are summarized in Table 2.

Fig. 8 shows the KCSIA\* jerk, acceleration and velocity profiles for  $x_1$  and  $x_2$  axes at different  $\zeta$  values. All profiles are within the pre-defined kinematic constraints (15). As  $\zeta \rightarrow 1$ ,  $T_{\text{cycle}} \rightarrow T_{\text{min}}$ , where at the time-optimal KCSIA\* has a shorter  $T_{\text{cycle}}$  than KCSIA. The KCSIA and KCSIA\* tangential velocities are depicted in Fig. 9(a). At both acute and obtuse corners, KCSIA has higher cornering velocities than KCSIA\* for  $\zeta \neq 1$ . This is a consequence of (34), where the upper bound cornering constraints are activated to maximize velocity as shown in Fig. 9(b). For KCSIA\*, the cornering velocities and  $\varepsilon_m$  depend on  $\zeta$ , where both parameters increase as  $\zeta \rightarrow 1$  and vice-versa. The correspondence between  $\varepsilon_m$  (Fig. 9(b)) and the generated cornering paths is shown in details 1 and 2 of Fig. 6.

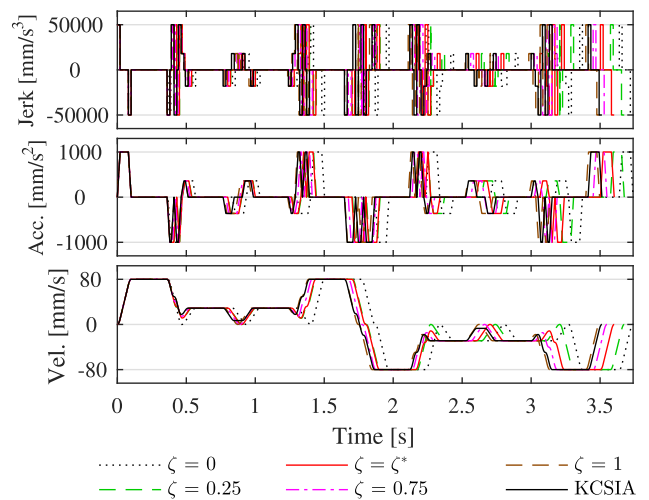
The computation time for generating the Pareto frontier (Fig. 7) at  $\delta_{\text{min}} = 0.02$  is 188.865 s, where it takes 0.861 s to compute the anchor points (7) and 188.865 s to perform the divide and conquer recursions. The average computation time per Pareto-optimal solution is 6.092 s. The choice of  $\delta_{\text{min}}$  affects computation time and the approximation of the best trade-off solution since it is a measure of Pareto frontier resolution. Table 3 shows influence of  $\delta_{\text{min}}$  on the best trade-off solution and computation time.

2) EXPERIMENTAL RESULTS

KCSIA\* has lower error peaks at acute cornering instances than KCSIA and KCSIA\* error tends to decrease as  $\zeta \rightarrow 0$



(a)  $x_1$  axis Pareto optimal trajectories



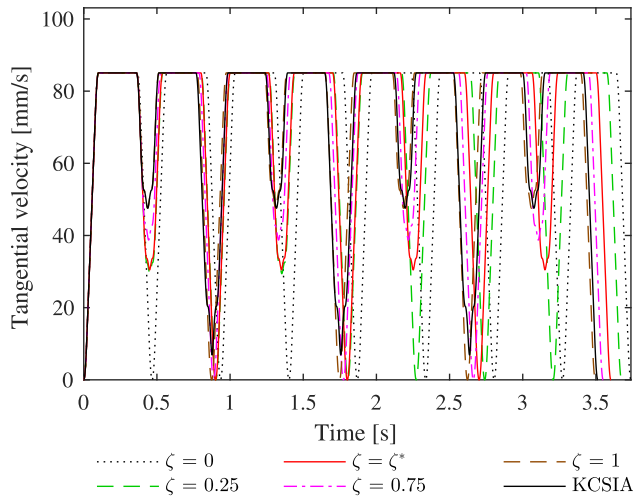
(b)  $x_2$  axis Pareto optimal trajectories

**FIGURE 8. An illustration of Pareto optimal jerk, acceleration and velocity profiles of the star-shaped tool path corresponding to different weighting factors  $\zeta$ , where kinematic constraints (15) are obeyed  $\forall \zeta$ .**

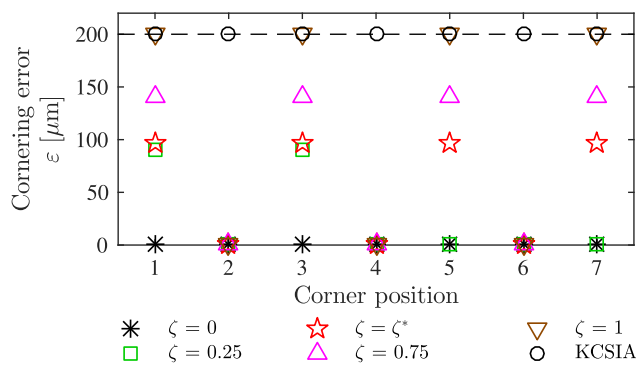
at obtuse corners (see Fig. 10). The KCSIA\* contouring errors are within the pre-defined cornering tolerance (i.e.,  $0 < \varepsilon \leq 200 \mu\text{m}$ ). The maximum and average contouring errors of each trajectory for all the trials are illustrated in Fig. 11, where it is shown that KCSIA\* has a better performance than KCSIA. Contouring error tends to increase as cycle time is reduced (i.e.,  $\zeta \rightarrow 1$ ). This is attributed to increased cornering velocities (see Fig. 9(a)).

The maximum contouring errors for  $\zeta = 0$  are non-zero minima  $\forall \zeta$  even though the cornering errors  $\varepsilon_m$  of the reference trajectory are approximately zero (Fig. 9(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 [55].

A  $1\sigma$  standard deviation is used to validate the contour error results, where Fig. 12 shows the consistency of KCSIA\*

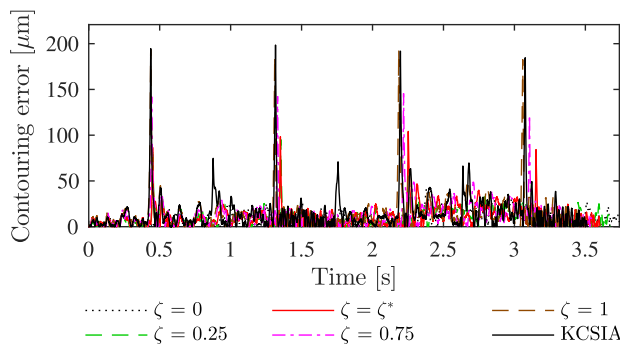


(a)



(b)

**FIGURE 9.** Tangential velocities and cornering errors of Pareto optimal trajectories for the star-shaped tool path at different weighting factors  $\zeta$ , where  $\epsilon$  is within the cornering constraints (16).

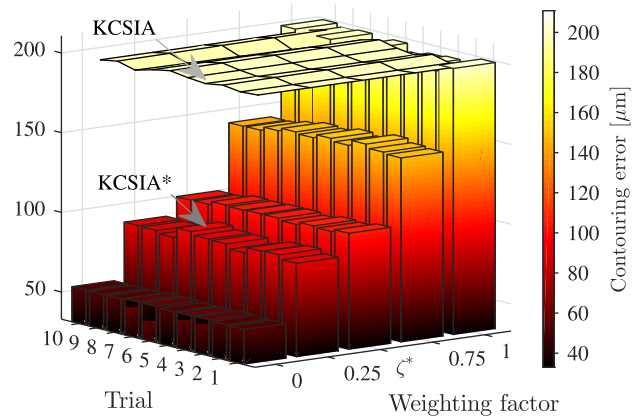


**FIGURE 10.** The 7<sup>th</sup> trial contouring error performance for the star-shaped tool path. The error tends to be within the set cornering tolerance  $0 < \epsilon \leq 200 \mu\text{m}$ .

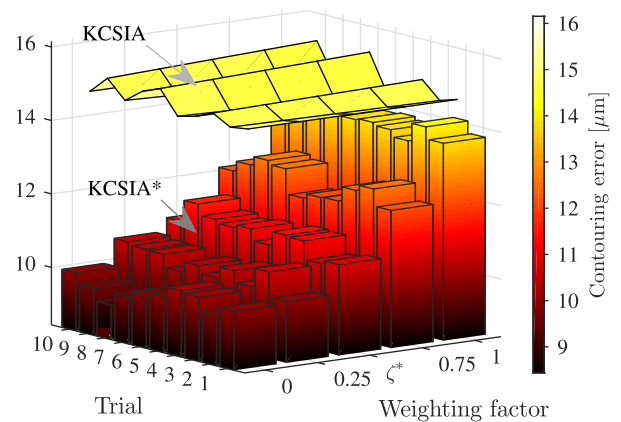
in performing better than KCSIA and the  $\zeta^*$  KCSIA\* trajectory having the best trade-off between cycle time and contouring error. Table 4 summarizes the experimental results of contouring performance.

**C. CASE II: COMPLEX TOOL PATH**

A relatively complicated tool path in the shape of a butterfly (Fig. 13) is used to study the effectiveness and limitations of

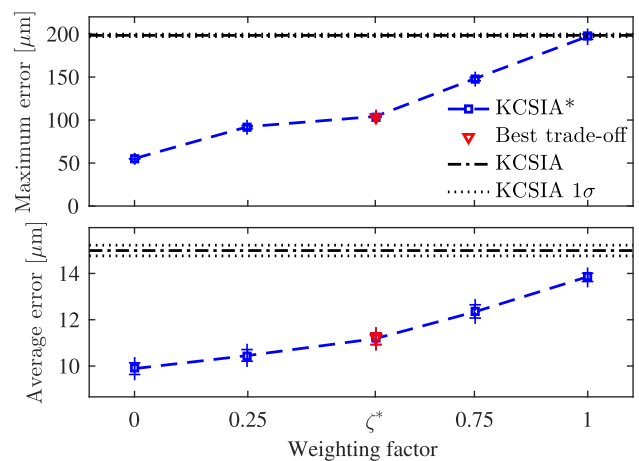


(a) Maximum error



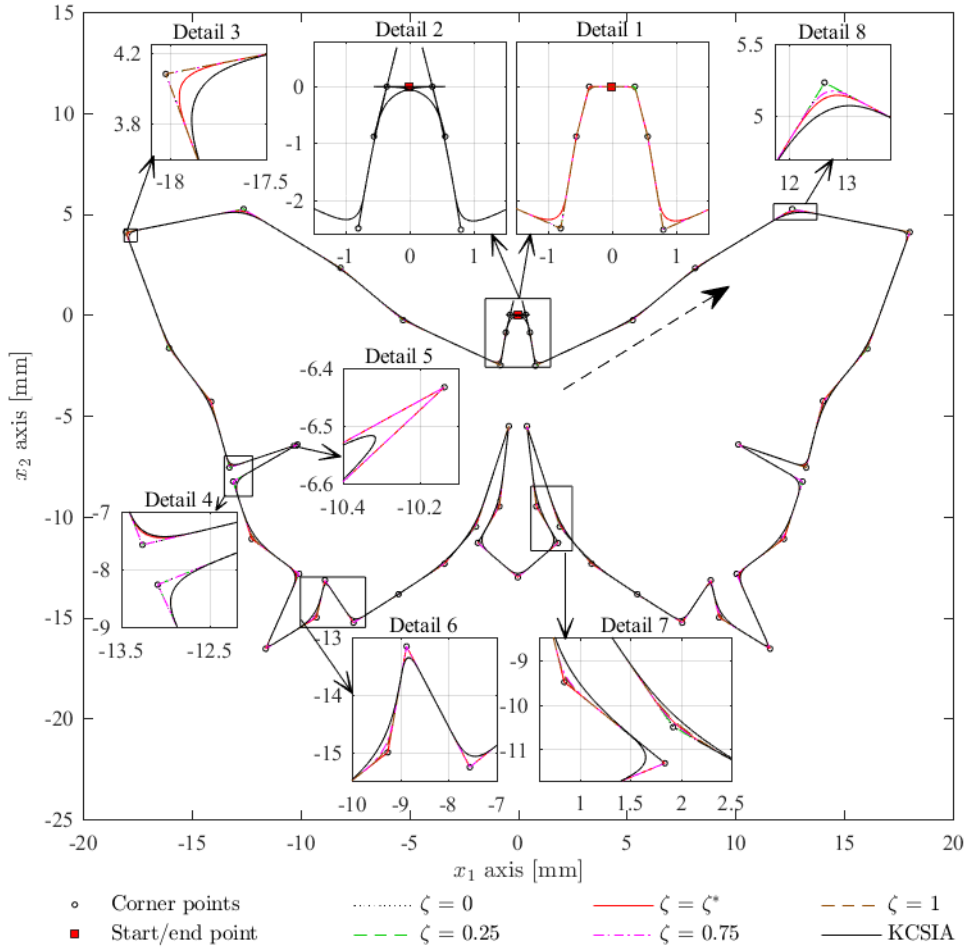
(b) Average error

**FIGURE 11.** Contouring performance experimental results for the star-shaped tool path.



**FIGURE 12.** Experimental contouring performance for the star-shaped tool path.

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.

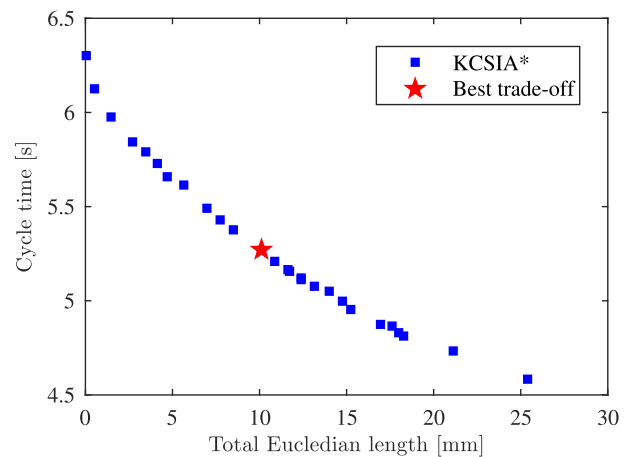


**FIGURE 13.** The butterfly-shaped path tool path used to represent complex paths in trajectory generation. The motion direction is shown by the dashed arrow.

1) OPTIMIZATION RESULTS

Pareto-optimal solutions of KCSIA\* are represented in Fig. 14. The KCSIA solution is infeasible since it consists of overlapping cornering paths as shown in detail 2 of Fig. 13. 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. 13, KCSIA\* avoids cornering path overlap by considering the geometric constraints (17). The best trade-off solution is located at  $\zeta^* = 0.5$ , where it reduces corner smoothing by 39.856 % while it increases cycle time by 15.022 % relative to the time-optimal KCSIA\* solution. A summary of the optimization results is shown in Table 5. At  $\delta_{min} = 0.02$ , the computation time required to generate the Pareto frontier (Fig. 14) 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  $\zeta$  values are within the specified kinematic constraints as shown in Fig. 15. The axial velocity constraints are not activated due to the proximity of corner points. Fig. 16

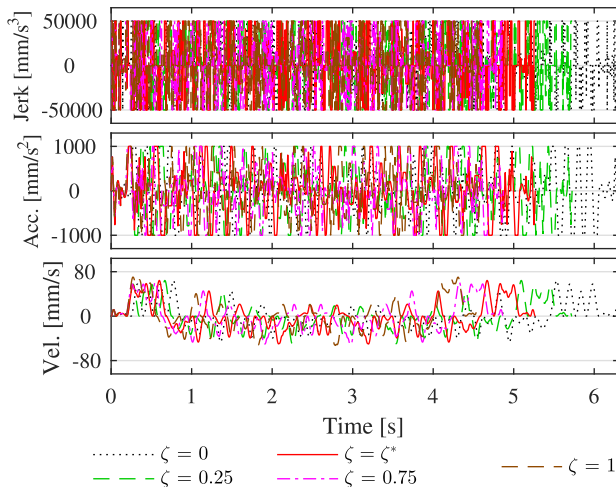


**FIGURE 14.** Optimization results for the complex tool path represented as a Pareto frontier.

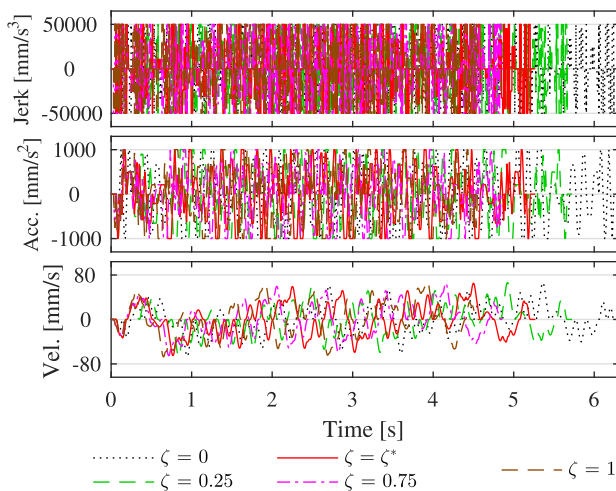
shows the extent of corner smoothing at different  $\zeta$  values. As  $\zeta \rightarrow 1$ , cornering errors tend to approach the upper bound of the tolerance in order to reduce cycle time.

TABLE 5. Results summary of the Pareto optimal solutions for the complex tool path.

Solution	Weighting factor $\zeta$	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	$1.421 \times 10^{-4}$	6.306	$\approx 100$	-37.686
KCSIA* : Best trade-off	$\zeta^* = 0.5$	10.117	5.268	39.856	-15.022



(a)  $x_1$  axis Pareto optimal trajectories



(b)  $x_2$  axis Pareto optimal trajectories

FIGURE 15. An illustration of Pareto-optimal jerk, acceleration and velocity profiles of the complex tool path corresponding to different weighting factors  $\zeta$ , where kinematic constraints (15) are obeyed  $\forall \zeta$ .

2) EXPERIMENTAL RESULTS

Contouring error profiles for different  $\zeta$  values are depicted in Fig. 17, where the maximum errors tend to be within the pre-defined cornering tolerance (i.e.,  $0 < \varepsilon \leq 200 \mu\text{m}$ ). Fig. 18 shows the maximum and average contouring errors of each trial. Similar to Section III-B2, the maximum contouring errors tend to decrease as  $\zeta \rightarrow 0$ , where non-zero minima

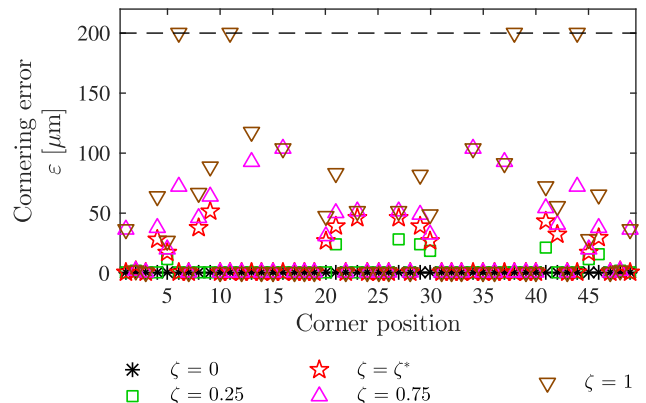


FIGURE 16. Cornering errors of Pareto-optimal trajectories of the complex tool path at different weighting factors  $\zeta$ , where  $\varepsilon$  is within the cornering constraints (16).

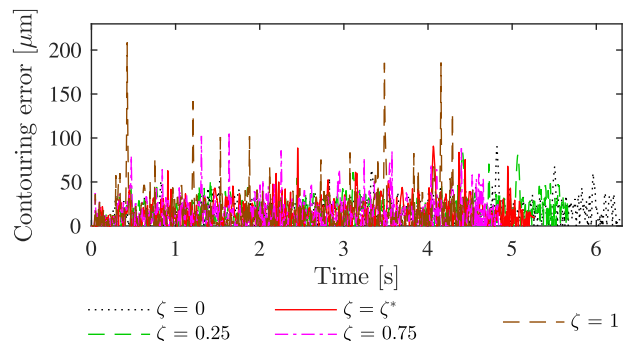


FIGURE 17. The 3<sup>rd</sup> trial contouring error performance for the complex tool path. The error tends to be within the set cornering tolerance  $0 < \varepsilon \leq 200 \mu\text{m}$ .

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. 18(b), where the average contouring errors increase as  $\zeta$  decreases beyond  $\zeta^*$ . Pareto-optimal trajectories which are generated at  $\zeta < \zeta^*$  have a higher number of corner points with  $\varepsilon_m \approx 0$  than those generated at  $\zeta \geq \zeta^*$  (Fig. 16).

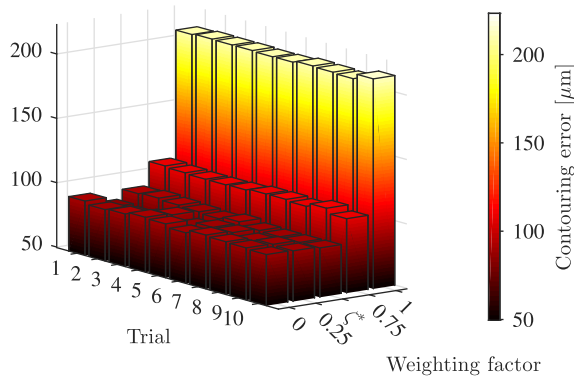
The repeatability in the maximum and average contouring error results is shown by the  $1\sigma$  standard deviations in Fig. 19. The best trade-off KCSIA\* trajectory shows the best compromise between cycle time and maximum/average contouring error. A summary of the experimental results is depicted in Table 6.

IV. DISCUSSION

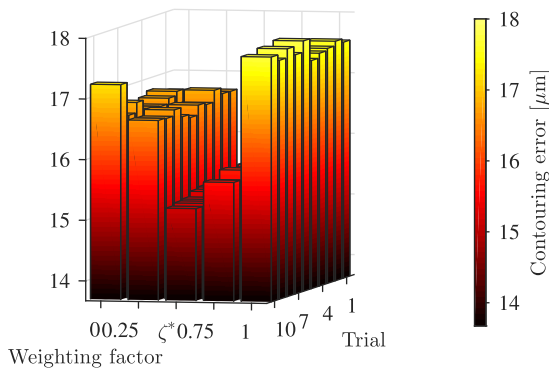
This study aims to solve the bi-objective problem of generating trajectories that minimize cycle time and maximize

TABLE 6. Experimental results summary of contouring performance for the complex tool path.

Trajectory	Weighting factor	Contouring error			
		Maximum	Average	Reduction relative to KCSIA*: $T_{cycle}$ optimal	
	$\zeta$	$\mu\text{m}$	$\mu\text{m}$	Maximum %	Average %
KCSIA*: $T_{cycle}$ optimal	1	209.07	17.62	-	-
KCSIA*: $L_{c,tot}$ optimal	0	90.53	16.76	56.70	4.87
KCSIA*: Best trade-off	$\zeta^* = 0.53125$	89.49	15.33	57.20	12.99



(a) Maximum error



(b) Average error

FIGURE 18. Contouring performance experimental results for the complex tool path.

motion accuracy for piecewise linear contours. The results indicate that the two objectives are contradictory, as illustrated in Figs. 12 and 19, where the best trade-off solutions are selected as the ones that are nearest to the unattainable utopia solutions. This study also depicts a correlation between corner smoothing and contouring error (Figs. 7 and 12 in Section III-B and Figs. 14 and 19 in Section III-C), where this result is in agreement with the findings of [34], [40].

Contrary to KLCS approaches proposed in [21], [33], [35], [36], [38], [39], [56], 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. 7 and 12. The proposed method can

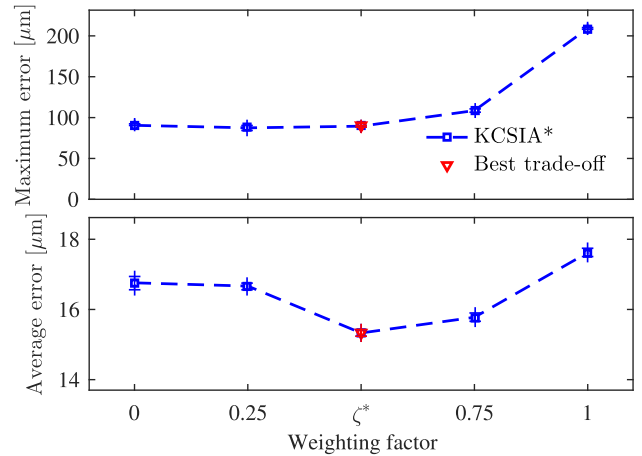


FIGURE 19. Experimental contouring performance for the complex tool path.

also avoid cornering path overlaps while maintaining Pareto-optimality (detail 1 in Fig. 13).

The results (Figs.12 and 19) signify that the proposed approach can be used to provide a trade-off between cycle time and motion accuracy within user-defined cornering tolerances. Although Pareto-optimal reference trajectories that favor accuracy can be generated, the minimum achievable contouring error finally depends on the feed drive system controller.

In contrast to [21], [33], [35], [36], [38], [39], [56], which have a  $\mathcal{O}(1)$  space complexity, the proposed method has a  $\mathcal{O}(n_l^2)$  space complexity, where  $n_l$  is the number of linear segments. This is because, in [21], [33], [35], [36], [38], [39], [56], motions are optimized along each corner and linear segment separately, while the proposed method optimizes motion for the entire trajectory. 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 III-C, there still exist many mechanical parts consisting of simple and moderated number of linear segments. Hence, the proposed method is effective for such parts.

V. CONCLUSION

This paper 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 bi-objective optimization problem formulated by the normalized normal constraint approach, where kinematic limits, continuity conditions and user-specified cornering tolerances 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 that is 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 47.53 % and 25.40 % while it increases cycle time by 2.53 %.

The proposed method is limited to symmetrical line-to-line corner transitions. Pareto-optimal corner smoothing with asymmetrical line-to-line, line-to-arc and arc-to-arc transitions is left as future works. The proposed method considers tool tip position in trajectory generation. In order to extend the method to 5- or 6-axis CNC machine tool applications, smooth transitions of tool orientation are required at corners. This consideration shall be done in future works. The proposed method requires significant computational resources for complex toolpaths with many corner points. Computational cost reduction is left as future works. The energy consumed by the generated Pareto-optimal was not scrutinized in this study. For future considerations, energy saving can be assimilated by extending the studied bi-objective problem to produce a Pareto surface of optimal solutions. Since the proposed method 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 [8]–[10] 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.

## APPENDIX

Algorithm 1 shows a pseudo code of the optimization process described in Section II. A candidate solution  $\psi_m$  is added into a Pareto set  $\Psi$  only if it passes removal [47] and significance criteria. For a 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

$$\hat{v}_i = \frac{1}{m_o - 1} E^{-1} P^{m_o - i} E \left[ v_1, \dots, v_{m_o - 1}, \sum_{i=1}^{m_o - 1} v_i \right]^T \quad (42)$$

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

are not positive, where  $E = \mathbb{1}_{m_o} - I_{m_o}$  and  $v_i$  is the  $i^{\text{th}}$  normalized normal constraint inequality constraint Lagrange multiplier.  $P$ ,  $\mathbb{1}_{m_o}$  and  $I_{m_o}$  are  $m_o \times m_o$  permutation, all-ones and identity matrices, respectively.

## Algorithm 1 Trajectory Optimization Algorithm

```

1: function trajectoryOptimizer( $p_{c,s}$ ,  $\varepsilon_{ub}$ ,  $j_{lim,k}$ ,  $a_{lim,k}$ ,
    $v_{lim,k}$ ,  $\delta_{min}$ )
2:   compute anchor points /* See (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:    $Z \leftarrow \{\zeta_l, \zeta_u\}$  /* Weighting factor set initializa-
   tion*/
   /* 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 (9) at  $\zeta = \zeta_m$ 
11:     $b_r \leftarrow$  removalCriterion( $\psi_m$ ) /* See (42)*/
12:     $b_s \leftarrow$  significanceCriterion( $\psi_m, \psi_l, \psi_u, \delta_{min}$ )
   /* See (43)*/
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:         $Z \leftarrow \{Z, \zeta_m\}$ 
17:         $\Psi \leftarrow$  sort( $\Psi$ ) /* Sort in ascending
   order of  $\zeta$ */
18:         $Z \leftarrow$  sort( $Z$ )
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  $Z$ 
28:      else
29:        break /* Terminate the divide and con-
   quer loop*/
30:      end if
31:    end if
32:  end while
33:   $\Psi \leftarrow$  paretoFilter( $\Psi$ ) /* From [45]*/
34:   $\psi^* \leftarrow$  compute (12)
35:   $\mu^* \leftarrow$  map  $\psi^*$  in the decision space
36:  return ( $\Psi, \psi^*, \mu^*$ )
37: end function

```

$\psi_m$ , with neighboring points  $\psi_l = [\psi_{l,1}, \psi_{l,2}]^T$  and  $\psi_u = [\psi_{u,1}, \psi_{u,2}]^T$ , is considered to be a significant Pareto-optimal point if

$$\min \left( |\psi_{l,1} - \psi_{m,1}|, |\psi_{l,2} - \psi_{m,2}|, |\psi_{u,1} - \psi_{m,1}|, |\psi_{u,2} - \psi_{m,2}| \right) \geq \delta_{min}, \quad (43)$$

where  $\delta_{\min}$  is a user-specified minimum trade-off level. The choice of  $\delta_{\min}$  affects the resolution of the generated Pareto frontier [42]. Since the removal and significance criteria cannot distinguish between local and global Pareto regions, a Pareto filter [45] is implemented to retain only global Pareto points once the divide and conquer loop is terminated.

## REFERENCES

- [1] T. V. Vardhan and B. S. Babu, "Multiaxis CNC programming and machining," in *Modern Manufacturing Processes* (Woodhead Publishing Reviews: Mechanical Engineering Series), K. Kumar and J. P. Davim, Eds. Sawston, U.K.: Woodhead Publishing, 2020, ch. 9, pp. 167–175.
- [2] Y. Altintas, A. Verl, C. Brecher, L. Uriarte, and G. Pritschow, "Machine tool feed drives," *CIRP Ann.*, vol. 60, no. 2, pp. 779–796, 2011, doi: [10.1016/j.cirp.2011.05.010](https://doi.org/10.1016/j.cirp.2011.05.010).
- [3] N. Uchiyama, "Adaptive two-degree-of-freedom control of feed drive systems," *Int. J. Mach. Tools Manuf.*, vol. 48, nos. 3–4, pp. 437–445, Mar. 2008, doi: [10.1016/j.ijmactools.2007.08.025](https://doi.org/10.1016/j.ijmactools.2007.08.025).
- [4] B. D. Bui, N. Uchiyama, and K. R. Simba, "Contouring control for three-axis machine tools based on nonlinear friction compensation for lead screws," *Int. J. Mach. Tools Manuf.*, vol. 108, pp. 95–105, Sep. 2016, doi: [10.1016/j.ijmactools.2016.06.001](https://doi.org/10.1016/j.ijmactools.2016.06.001).
- [5] M. R. Msukwa, E. W. Nshama, and N. Uchiyama, "Contouring control based on reference adjustment and uncertainty compensator for precision motion of industrial feed drive systems," in *Proc. IEEE 29th Int. Symp. Ind. Electron.*, Jun. 2020, pp. 89–94, doi: [10.1109/ISIE45063.2020.9152478](https://doi.org/10.1109/ISIE45063.2020.9152478).
- [6] H. Chen and N. Sun, "Nonlinear control of underactuated systems subject to both actuated and unactuated state constraints with experimental verification," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 7702–7714, Sep. 2020, doi: [10.1109/TIE.2019.2946541](https://doi.org/10.1109/TIE.2019.2946541).
- [7] H. Chen and N. Sun, "An output feedback approach for regulation of 5-DOF offshore cranes with ship yaw and roll perturbations," *IEEE Trans. Ind. Electron.*, early access, Feb. 2, 2021, doi: [10.1109/TIE.2021.3055159](https://doi.org/10.1109/TIE.2021.3055159).
- [8] Y. M. Hendrawan, K. R. Simba, and N. Uchiyama, "Iterative learning based trajectory generation for machine tool feed drive systems," *Robot. Comput.-Integr. Manuf.*, vol. 51, pp. 230–237, Jun. 2018, doi: [10.1016/j.rcim.2017.12.009](https://doi.org/10.1016/j.rcim.2017.12.009).
- [9] Y.-H. Lee, S.-C. Hsu, T.-Y. Chi, Y.-Y. Du, J.-S. Hu, and T.-C. Tsao, "Industrial robot accurate trajectory generation by nested loop iterative learning control," *Mechatronics*, vol. 74, Apr. 2021, Art. no. 102487, doi: [10.1016/j.mechatronics.2021.102487](https://doi.org/10.1016/j.mechatronics.2021.102487).
- [10] B. Li, H. Zhang, P. Ye, and J. Wang, "Trajectory smoothing method using reinforcement learning for computer numerical control machine tools," *Robot. Comput.-Integr. Manuf.*, vol. 61, Feb. 2020, Art. no. 101847, doi: [10.1016/j.rcim.2019.101847](https://doi.org/10.1016/j.rcim.2019.101847).
- [11] N. Uchiyama, K. Mori, K. Terashima, T. Saeki, T. Kamigaki, and H. Kawamura, "Optimal motion trajectory generation and real-time trajectory modification for an industrial robot working in a rectangular space," *J. Syst. Design Dyn.*, vol. 7, no. 3, pp. 278–292, 2013, doi: [10.1299/jstd.7.278](https://doi.org/10.1299/jstd.7.278).
- [12] B. Sencer, Y. Altintas, and E. Croft, "Feed optimization for five-axis CNC machine tools with drive constraints," *Int. J. Mach. Tools Manuf.*, vol. 48, nos. 7–8, pp. 733–745, Jun. 2008, doi: [10.1016/j.ijmactools.2008.01.002](https://doi.org/10.1016/j.ijmactools.2008.01.002).
- [13] B. Sencer and S. Tajima, "Frequency optimal feed motion planning in computer numerical controlled machine tools for vibration avoidance," *J. Manuf. Sci. Eng.*, vol. 139, no. 1, Jan. 2017, Art. no. 011006, doi: [10.1115/1.4034140](https://doi.org/10.1115/1.4034140).
- [14] S. Kucuk, "Optimal trajectory generation algorithm for serial and parallel manipulators," *Robot. Comput.-Integr. Manuf.*, vol. 48, pp. 219–232, Dec. 2017, doi: [10.1016/j.rcim.2017.04.006](https://doi.org/10.1016/j.rcim.2017.04.006).
- [15] Y. Fang, J. Qi, J. Hu, W. Wang, and Y. Peng, "An approach for jerk-continuous trajectory generation of robotic manipulators with kinematical constraints," *Mech. Mach. Theory*, vol. 153, Nov. 2020, Art. no. 103957, doi: [10.1016/j.mechmachtheory.2020.103957](https://doi.org/10.1016/j.mechmachtheory.2020.103957).
- [16] Z. Wang, X. Zhang, L. He, and T. Liu, "Model predictive contouring control for biaxial feed drive system," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Aug. 2020, pp. 5237–5242, doi: [10.1109/CCDC49329.2020.9163999](https://doi.org/10.1109/CCDC49329.2020.9163999).
- [17] R. Teunter, O. Tang, and K. Kaparis, "Heuristics for the economic lot scheduling problem with returns," *Int. J. Prod. Econ.*, vol. 118, no. 1, pp. 323–330, Mar. 2009, doi: [10.1016/j.ijpe.2008.08.036](https://doi.org/10.1016/j.ijpe.2008.08.036).
- [18] M. Mohammadi, S. N. Musa, and A. Bahreinejad, "Optimization of mixed integer nonlinear economic lot scheduling problem with multiple setups and shelf life using metaheuristic algorithms," *Adv. Eng. Softw.*, vol. 78, pp. 41–51, Dec. 2014, doi: [10.1016/j.advengsoft.2014.08.004](https://doi.org/10.1016/j.advengsoft.2014.08.004).
- [19] S. Y. Jeong, Y. J. Choi, P. Park, and S. G. Choi, "Jerk limited velocity profile generation for high speed industrial robot trajectories," *IFAC Proc. Volumes*, vol. 38, no. 1, pp. 595–600, 2005, doi: [10.3182/20050703-6-cz-1902.01369](https://doi.org/10.3182/20050703-6-cz-1902.01369).
- [20] P. Besset and R. Béarée, "FIR filter-based online jerk-constrained trajectory generation," *Control Eng. Pract.*, vol. 66, pp. 169–180, Sep. 2017, doi: [10.1016/j.conengprac.2017.06.015](https://doi.org/10.1016/j.conengprac.2017.06.015).
- [21] W. Wang, C. Hu, K. Zhou, S. He, and L. Zhu, "Local asymmetrical corner trajectory smoothing with bidirectional planning and adjusting algorithm for CNC machining," *Robot. Comput.-Integr. Manuf.*, vol. 68, Apr. 2021, Art. no. 102058, doi: [10.1016/j.rcim.2020.102058](https://doi.org/10.1016/j.rcim.2020.102058).
- [22] S. Tajima and B. Sencer, "Global tool-path smoothing for CNC machine tools with uninterrupted acceleration," *Int. J. Mach. Tools Manuf.*, vol. 121, pp. 81–95, Oct. 2017, doi: [10.1016/j.ijmactools.2017.03.002](https://doi.org/10.1016/j.ijmactools.2017.03.002).
- [23] Q.-B. Xiao, M. Wan, X.-B. Qin, Y. Liu, and W.-H. Zhang, "Real-time smoothing of G01 commands for five-axis machining by constructing an entire spline with the bounded smoothing error," *Mech. Mach. Theory*, vol. 161, Jul. 2021, Art. no. 104307, doi: [10.1016/j.mechmachtheory.2021.104307](https://doi.org/10.1016/j.mechmachtheory.2021.104307).
- [24] D.-N. Song, J.-W. Ma, Y.-G. Zhong, and J.-J. Yao, "Global smoothing of short line segment toolpaths by control-point-assigning-based geometric smoothing and FIR filtering-based motion smoothing," *Mech. Syst. Signal Process.*, vol. 160, Nov. 2021, Art. no. 107908, doi: [10.1016/j.ymsp.2021.107908](https://doi.org/10.1016/j.ymsp.2021.107908).
- [25] M. Wan, X.-B. Qin, Q.-B. Xiao, Y. Liu, and W.-H. Zhang, "Asymmetrical pythagorean-hodograph (PH) spline-based C<sup>3</sup> continuous corner smoothing algorithm for five-axis tool paths with short segments," *J. Manuf. Processes*, vol. 64, pp. 1387–1411, Apr. 2021, doi: [10.1016/j.jmapro.2021.02.059](https://doi.org/10.1016/j.jmapro.2021.02.059).
- [26] K. Erkorkmaz and Y. Altintas, "Quintic spline interpolation with minimal feed fluctuation," *J. Manuf. Sci. Eng.*, vol. 127, no. 2, pp. 339–349, Apr. 2005, doi: [10.1115/1.1830493](https://doi.org/10.1115/1.1830493).
- [27] B. Sencer, K. Ishizaki, and E. Shamoto, "A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of NC systems along linear tool paths," *Int. J. Adv. Manuf. Technol.*, vol. 76, nos. 9–12, pp. 1977–1992, 2015, doi: [10.1007/s00170-014-6386-2](https://doi.org/10.1007/s00170-014-6386-2).
- [28] W. Wang, C. Hu, K. Zhou, and S. He, "(B,6)Corner trajectory smoothing with asymmetrical transition profile for CNC machine tools," *Int. J. Mach. Tools Manuf.*, vol. 144, Sep. 2019, Art. no. 103423, doi: [10.1016/j.ijmactools.2019.05.007](https://doi.org/10.1016/j.ijmactools.2019.05.007).
- [29] Y. Yan, L. Zhang, and K. Zhang, "Corner smoothing transition algorithm for five-axis linear tool path," *Procedia CIRP*, vol. 56, pp. 604–609, Jan. 2016, doi: [10.1016/j.procir.2016.10.119](https://doi.org/10.1016/j.procir.2016.10.119).
- [30] J. Yang and A. Yuen, "An analytical local corner smoothing algorithm for five-axis CNC machining," *Int. J. Mach. Tools Manuf.*, vol. 123, pp. 22–35, Dec. 2017, doi: [10.1016/j.ijmactools.2017.07.007](https://doi.org/10.1016/j.ijmactools.2017.07.007).
- [31] J. Huang, X. Du, and L.-M. Zhu, "Real-time local smoothing for five-axis linear toolpath considering smoothing error constraints," *Int. J. Mach. Tools Manuf.*, vol. 124, pp. 67–79, Jan. 2018, doi: [10.1016/j.ijmactools.2017.10.001](https://doi.org/10.1016/j.ijmactools.2017.10.001).
- [32] J. Shi, Q. Bi, L. Zhu, and Y. Wang, "Corner rounding of linear five-axis tool path by dual PH curves blending," *Int. J. Mach. Tools Manuf.*, vol. 88, pp. 223–236, Jan. 2015, doi: [10.1016/j.ijmactools.2014.09.007](https://doi.org/10.1016/j.ijmactools.2014.09.007).
- [33] S. Tajima and B. Sencer, "Kinematic corner smoothing for high speed machine tools," *Int. J. Mach. Tools Manuf.*, vol. 108, pp. 27–43, Sep. 2016, doi: [10.1016/j.ijmactools.2016.05.009](https://doi.org/10.1016/j.ijmactools.2016.05.009).
- [34] E. W. Nshama, M. R. Msukwa, and N. Uchiyama, "A trade-off between energy saving and cycle time reduction by Pareto optimal corner smoothing in industrial feed drive systems," *IEEE Access*, vol. 9, pp. 23579–23594, 2021, doi: [10.1109/ACCESS.2021.3056755](https://doi.org/10.1109/ACCESS.2021.3056755).
- [35] B. Sencer, K. Ishizaki, and E. Shamoto, "High speed cornering strategy with confined contour error and vibration suppression for CNC machine tools," *CIRP Ann.*, vol. 64, no. 1, pp. 369–372, 2015, doi: [10.1016/j.cirp.2015.04.102](https://doi.org/10.1016/j.cirp.2015.04.102).
- [36] B. Sencer, Y. Kakinuma, and Y. Yamada, "Linear interpolation of machining tool-paths with robust vibration avoidance and contouring error control," *Precis. Eng.*, vol. 66, pp. 269–281, Nov. 2020, doi: [10.1016/j.precisioneng.2020.04.007](https://doi.org/10.1016/j.precisioneng.2020.04.007).

- [37] A. Shahzadeh, A. Khosravi, S. Nahavandi, and T. Robinette, "Smooth path planning using biclothoid fillets for high speed CNC machines," *Int. J. Mach. Tools Manuf.*, vol. 132, pp. 36–49, Sep. 2018, doi: [10.1016/j.ijmactools.2018.04.003](https://doi.org/10.1016/j.ijmactools.2018.04.003).
- [38] Q.-B. Xiao, M. Wan, Y. Liu, X.-B. Qin, and W.-H. Zhang, "Space corner smoothing of CNC machine tools through developing 3D general clothoid," *Robot. Comput.-Integr. Manuf.*, vol. 64, Aug. 2020, Art. no. 101949, doi: [10.1016/j.rcim.2020.101949](https://doi.org/10.1016/j.rcim.2020.101949).
- [39] X. Huang, F. Zhao, T. Tao, and X. Mei, "A newly developed corner smoothing methodology based on clothoid splines for high speed machine tools," *Robot. Comput.-Integr. Manuf.*, vol. 70, Aug. 2021, Art. no. 102106, doi: [10.1016/j.rcim.2020.102106](https://doi.org/10.1016/j.rcim.2020.102106).
- [40] K. Erkorkmaz, C.-H. Yeung, and Y. Altintas, "Virtual CNC system. Part II. High speed contouring application," *Int. J. Mach. Tools Manuf.*, vol. 46, no. 10, pp. 1124–1138, Aug. 2006, doi: [10.1016/j.ijmactools.2005.08.001](https://doi.org/10.1016/j.ijmactools.2005.08.001).
- [41] F. Logist, B. Houska, M. Diehl, and J. Van Impe, "Fast Pareto set generation for nonlinear optimal control problems with multiple objectives," *Struct. Multidisciplinary Optim.*, vol. 42, no. 4, pp. 591–603, 2010, doi: [10.1007/s00158-010-0506-x](https://doi.org/10.1007/s00158-010-0506-x).
- [42] I. Hashem, D. Telen, P. Nimmegheers, F. Logist, and J. Van Impe, "A novel algorithm for fast representation of a Pareto front with adaptive resolution: Application to multi-objective optimization of a chemical reactor," *Comput. Chem. Eng.*, vol. 106, pp. 544–558, Nov. 2017, doi: [10.1016/j.compchemeng.2017.06.020](https://doi.org/10.1016/j.compchemeng.2017.06.020).
- [43] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," *Struct. Optim.*, vol. 14, no. 1, pp. 63–69, Aug. 1997.
- [44] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.
- [45] A. Messac, A. Ismail-Yahaya, and C. A. Mattson, "The normalized normal constraint method for generating the Pareto frontier," *Struct. Multidisciplinary Optim.*, vol. 25, no. 2, pp. 86–98, 2003, doi: [10.1007/s00158-002-0276-1](https://doi.org/10.1007/s00158-002-0276-1).
- [46] J. Sanchis, M. Martínez, X. Blasco, and J. V. Salcedo, "A new perspective on multiobjective optimization by enhanced normalized normal constraint method," *Struct. Multidisciplinary Optim.*, vol. 36, no. 5, pp. 537–546, 2008.
- [47] F. Logist and J. Van Impe, "Novel insights for multi-objective optimisation in engineering using normal boundary intersection and (enhanced) normalised normal constraint," *Struct. Multidisciplinary Optim.*, vol. 45, no. 3, pp. 417–431, Mar. 2012, doi: [10.1007/s00158-011-0698-8](https://doi.org/10.1007/s00158-011-0698-8).
- [48] I. Y. Kim and O. L. de Weck, "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation," *Struct. Multidisciplinary Optim.*, vol. 29, no. 2, pp. 149–158, 2005, doi: [10.1007/s00158-004-0465-1](https://doi.org/10.1007/s00158-004-0465-1).
- [49] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. Part I: Jerk limited trajectory generation and quintic spline interpolation," *Int. J. Mach. Tools Manuf.*, vol. 41, no. 9, pp. 1323–1345, 2001, doi: [10.1016/s0890-6955\(01\)00002-5](https://doi.org/10.1016/s0890-6955(01)00002-5).
- [50] P. E. Gill and E. Wong, "Sequential quadratic programming methods," in *Mixed Integer Nonlinear Programming*, J. Lee and S. Leyffer, Eds. New York, NY, USA: Springer, 2012, pp. 147–224.
- [51] O. Goldreich, "Computational complexity: A conceptual perspective," *ACM SIGACT News*, vol. 39, pp. 35–39, 2008.
- [52] K. Schittkowski, "An active set strategy for solving optimization problems with up to 200,000,000 nonlinear constraints," *Appl. Numer. Math.*, vol. 59, no. 12, pp. 2999–3007, Dec. 2009, doi: [10.1016/j.apnum.2009.07.009](https://doi.org/10.1016/j.apnum.2009.07.009).
- [53] M. R. Msukwa and N. Uchiyama, "Design and experimental verification of adaptive sliding mode control for precision motion and energy saving in feed drive systems," *IEEE Access*, vol. 7, pp. 20178–20186, 2019, doi: [10.1109/ACCESS.2019.2897844](https://doi.org/10.1109/ACCESS.2019.2897844).
- [54] M. R. Msukwa, E. W. Nshama, and N. Uchiyama, "Adaptive sliding mode control with feedforward compensator for energy-efficient and high-speed precision motion of feed drive systems," *IEEE Access*, vol. 8, pp. 43571–43581, 2020, doi: [10.1109/ACCESS.2020.2977395](https://doi.org/10.1109/ACCESS.2020.2977395).
- [55] K. Johansson and C. Canudas-de-Wit, "Revisiting the LuGre friction model," *IEEE Control Syst. Mag.*, vol. 28, no. 6, pp. 101–114, Dec. 2008, doi: [10.1109/MCS.2008.929425](https://doi.org/10.1109/MCS.2008.929425).
- [56] S. Tajima and B. Sencer, "Real-time trajectory generation for 5-axis machine tools with singularity avoidance," *CIRP Ann.*, vol. 69, no. 1, pp. 349–352, 2020, doi: [10.1016/j.cirp.2020.04.050](https://doi.org/10.1016/j.cirp.2020.04.050).



and mechatronics systems design and control.

**ENOCK WILLIAM NSHAMA** received the B.Sc. degree in electromechanical engineering from the University of Dar es Salaam, Tanzania, in 2014, and the M.E. degree from Toyohashi University of Technology, Japan, in 2018, where he is currently pursuing the Ph.D. degree. Since 2016, he has been with the Department of Mechanical Engineering, University of Dar es Salaam, where he is currently an Assistant Lecturer. His research interests include machine tool control, industrial robotics,



of Mechanical Engineering, Toyohashi University of Technology, Aichi, Japan, where he is currently a Professor. He is a member of IEEE IES, CSS, and RAS.

**NAOKI UCHIYAMA** (Member, IEEE) received the B.E. degree from Numazu National College of Technology, Shizuoka, Japan, in 1988, and the B.E. and M.E. degrees from Shizuoka University, Shizuoka, Japan, in 1990 and 1992, respectively, and the Ph.D. degree in mechanical engineering from Tokyo Metropolitan University, Tokyo, Japan, in 1995. He was a Visiting Scholar with the University of California at Davis, from 2001 to 2002. Since 1995, he has been with the Department

...