MM Science Journal | www.mmscience.eu
ISSN 1803-1269 (Print) | ISSN 1805-0476 (On-line)
Special Issue | HSM 2025
18th International Conference on High Speed Machining
October 15-16, 2025, Metz, France
DOI: 10.17973/MMSJ.2025_11_2025140

HSM2025-45142

DIGITAL THREAD ENABLED TIME OPTIMAL TRAJECTORY GENERATION FOR MACHINING TOOLPATHS

D. Wilkinson[1,2]*, B. Sencer[3], R. Ward[1,2]

[1]University of Sheffield, School of Electrical and Electronic Engineering, Sheffield, United Kingdom

[2]University of Sheffield, Advanced Manufacturing Research Centre, Sheffield, United Kingdom

[3]Oregon State University, Department of Mechanical, Industrial, and Manufacturing Engineering, Corvallis, Oregon, United States of America

*Corresponding author; e-mail: dwilkinson2@sheffield.ac.uk

Abstract
The manufacturing sector is undergoing a data-driven revolution, with standards such as ISO 10303 (STEP) enabling a seamless digital thread – enhancing productivity, sustainability, and regulatory compliance. As the complexity of manufacturing operations grows, so does the demand for end-to-end certifiable processes via interoperable, standards-based digital workflows to maintain traceability and trust – from design, through to manufacture and beyond. This study introduces a novel ISO 10303-238 (STEP-NC) compatible CNC toolpath trajectory generation framework. Uniquely, it allows feature-based customisation in the CNC interpolation stage, generating minimum time toolpaths within geometric tolerances and machine limits. The proposed method has been validated against an industrial controller, showing cycle time reductions up to 14.37%.

Keywords:
CNC machining, Trajectory generation, Numerical control, Digital thread

1 INTRODUCTION

With the ever increasing interest in sustainable, responsible, and analytics-driven processes [Raoufi 2024], the manufacturing industry is rapidly transitioning towards traceable and interoperable standards-based practices, supported by a breadth of research developing both the physical and digital infrastructure to support this transition [Imad 2022, Yuan 2022]. The digital thread is the underpinning framework that establishes data continuity and traceability across the entire lifecycle of a product [Abdel-Aty 2024], with adoption of the digital thread opening up the opportunity for energy-efficient manufacturing [Cheng 2022]. The introduction of the STEP standard (ISO 10303 [[International Organization for Standardization, 2019]) paved the way for standardised, implementation-agnostic file formats for feature-driven design and engineering, allowing seamless data flow from the computer aided design (CAD) stage through to computer aided manufacture (CAM) [Liu 2018].

Traditionally, computer numerical control (CNC) machine tools have been utilised as a standalone process in the CAD-CAM-CNC cycle. This is evident when observing the G-code standard (ISO 6983-1 [International Organization for Standardization 2009]), which standardises the numerical control code format used by CNC machines globally. G-code is generated at the CAM stage, and then postprocessed, wherein machine-specific information (datums, plane shifts, etc.) are added to the tool centre point (TCP) and tool orientation vector (TOV) coordinates. Aside from feeds, speeds, and TCP/TOV tolerances, there is lack of further contextual information to support data continuity from CAD/CAM to CNC [Liu 2018]. STEP-NC (ISO 10303-238 [International Organization for Standardization 2022]) has been developed to overcome this through extending the STEP format to allow intelligent data-enabled control of machine tools. The STEP-NC standard allows greater richness of geometric dimensioning and tolerancing (GD&T) information into the machining stage - a key limitation of conventional G-code based machine control.

In the scope of subtractive manufacturing, the digital thread is still to reach technological maturity, with only a small number of original equipment manufacturers (OEMs) developing STEP-enabled technology [Suh 2003]. Fundamentally, STEP-enabled machine tools do not function on the traditional digital workflow required for processing G-code numerical control (NC) programs [Rauch 2012], in which the NC kernel (NCK) parses and interpolates G-code, generating continuous axial reference trajectories for the machine tool feed drive systems. For an in-depth discussion on the differences in system architecture for STEP-compliant CNC, along with the future of such systems, one should familiarise themselves with Nguyen [Nguyen 2009]. The work of Rauch was fundamental in advancing the STEP-NC standard [Rauch

2012], and showcasing how existing CNC controllers can be adapted to conform to the STEP-NC standard.

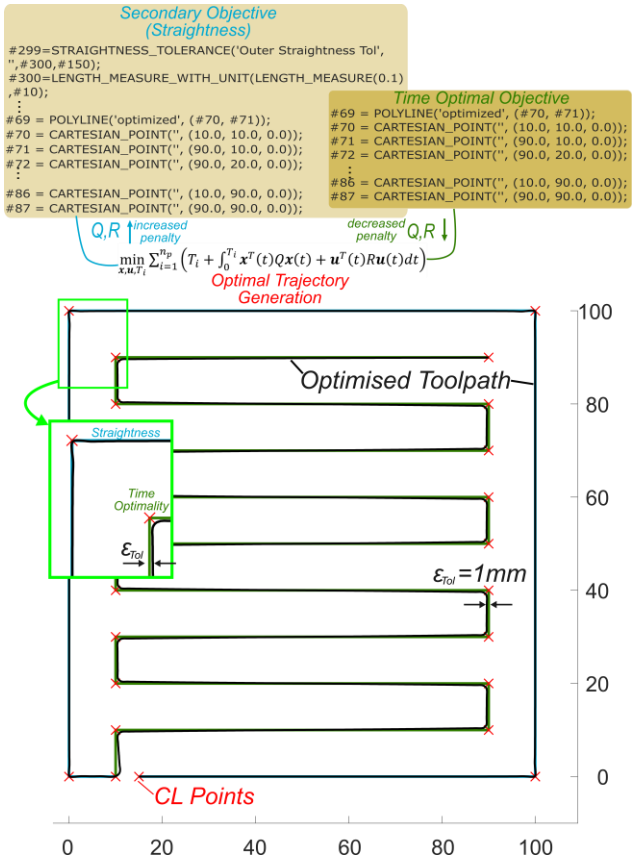

Fig. 1: Digital thread enabled optimal trajectory generation, including an example STEP-NC file excerpt with geometric tolerancing information.

Parsing geometric information from a source file is the first step taken by the NCK. The subsequent stage involves generating a toolpath trajectory that interpolates on the given geometric coordinates and produces a set of continuous motion reference signals, ready to be passed to the machine's axial feed drive control system [Altintas 2012]. Methods on the generation of the axial motion control trajectories can be grouped into two main approaches, being (i) geometrical interpolation methods using splines and (ii) digital signal processing-based smoothing using finite impulse response (FIR) filters [Yan 2023b]. FIR interpolation is a computationally efficient method of generating kinematically constrained feed drive reference signals [Ward 2022], and has been successfully applied in both local and global smoothing of toolpaths [Tajima 2022], with methods developed in constraining contour error and orientation error in both 3-axis and 5-axis toolpaths [Yan 2023b]. Spline interpolation, on the other hand, is a geometric approach to toolpath smoothing, which aims to model toolpaths as piecewise continuous parametric curves to high degrees of geometric and parametric continuity [Yan 2023a], enabling jerk limited motion profiles [Erkorkmaz 2001, Hu 2023]. Spline-based smoothing methods typically allow higher degrees of flexibility in the resultant toolpath motion at the expense of computational complexity [Wang 2025].

Whilst both approaches have key benefits in terms of computation and/or ease of implementation, there is limited ability for inclusion of time optimality in either approach.

To solve this, recent research has focused on posing toolpath trajectory generation as an optimisation problem. Bosetti and Ragni proposed a minimum time toolpath trajectory generation method using optimal control theory [Bosetti 2016], through first modelling the interpolator as a dynamical system in the displacement domain. This method allowed constraining of path error and derivatives of motion up to jerk and was successful in reducing part program execution (cycle) time, with reductions in cycle time of up to 20% compared to an industrial controller. However, this research was limited to local corner smoothing.

Lin et al. then proposed a time optimal trajectory generation method for the sequential waypoint tracking problem [Lin 2018] utilising MPC to reduce the computational complexity of the trajectory generation problem. Positional errors along each linear motion command were constrained through a nonconvex cross product constraint. With such approach, there are no guarantees of global optimality, for one cannot compute the numerical derivative of this nonconvex constraint.

Zhang developed a convex quadratic programming based MPC algorithm to optimise the axial drive motion profiles based on a model of the drive system [Zhang 2019]. The method allowed constraining of motion derivatives up to axial jerk. Despite successful verification in simulation, the formulation of the TCP error constraints affects the ease of implementation of the proposed algorithm, particularly when scaling the method to 3-axis toolpaths and beyond.

The use of digital twins in the feedrate optimisation problem has also been investigated, with Kim proposing an uncertainty-aware digital twin model for feedrate optimisation [Kim 2024]. This research built on prior methods which combined feed drive error pre-compensation with feedrate optimisation [Kim 2020], allowing incorporation of machine uncertainties stemming from both actuation behaviour and controller performance. By solely focusing on feedrate optimisation along a toolpath parametrised in the displacement domain, it was shown that linear programming may be used to solve for a time optimal error constrained feedrate profile, hence reducing the computational cost of the optimisation stage.

Feedrate optimisation methods formulated in the displacement domain, like in Kim are compatible with STEP-NC, with the standard allowing for description of CNC toolpaths in parametrised curve form [International Organization for Standardization 2021]. However, when a precomputed parametrised toolpath is not available, then the NC interpolator must perform both feedrate and trajectory optimisation, based on the NC part program. In such cases, STEP-NC, much like conventional G-code, still suffers from the problem of optimal trajectory generation given a simple sequence of cutter location (CL) points. This problem is therefore still prevalent in STEP-NC files with **polyline** curve definitions.

In this work, we aim to overcome these issues. Our contributions are as follows: (i) development of a novel optimisation-based toolpath trajectory generation algorithm capable of global smoothing, and (ii) proposal of a framework for incorporating GD&T data into the CNC trajectory generation stage through STEP-NC modification. The proposed algorithm has been applied to multiple industrial toolpaths and benchmarked against a commercially available machine tool.

2 STEP-NC INTEGRATION

This section will describe STEP-NC enabled optimal trajectory generation and discuss a mechanism of passing geometrical information (including dimensioning and tolerancing) to the numerical control interpolator function.

STEP-NC files offer greater customisability in the part program, allowing feature-based manufacturing. The increased customisability with STEP-NC files can be exploited to provide further information to the NCK. Taking the standard **polyline** command, one may enrich a simple sequence of linear motion commands with a feature-specific geometric constraint such as straightness on an exterior geometry (see Fig. 1), or a cylindricity tolerance on a suitable three-dimensional geometry.

With advancements in multi-objective optimisation-based CNC interpolation, there must exist means of providing further imperatives to the CNC interpolator – aside from tolerances. Here, we utilise a simple standardised method that exploits the existing STEP-NC format: enriching the **bounded_curve** motion commands (such as the **polyline** command) with a tag such as **polyline('optimized, points[])**, thereby informing the optimal interpolator to generate a minimum time toolpath based on the provided contextual information (i.e., GD&T tolerances).

3 OPTIMAL TRAJECTORY GENERATION

In this section, we introduce the toolpath trajectory generation process as a constrained optimisation problem.

Toolpaths are defined as a sequence of discrete coordinates, either in (x, y, z) or (x, y, z, i, j, k) form for 3- and 5-axis machining, respectively. The objective of toolpath trajectory generation is to interpolate a minimum time continuous tool-tip motion profile along the sequence of coordinates (CL-data points), whilst conforming to both geometrical tolerances in the tool centre point (TCP) and tool orientation vector (TOV) sense and kinematic (axial velocity and acceleration) limits.

In addition, toolpaths should be designed to avoid excitation of the machine's resonant modes. This is usually achieved through either filtering the motion commands using a tuned filter [Tajima 2018], or constraining axial and/or tangential jerk [Erkorkmaz 2001]. In this work, we will focus on generating jerk limited 2D planar toolpaths. The proposed algorithm has been successfully implemented in 3-axis toolpaths, however, for brevity, only a 2-axis formulation is introduced in this work. The interpolator can be considered as a continuous time dynamical system,

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t), \tag{1}$$

with the system matrices A and B defined as

$$\begin{bmatrix} \dot{s}_x(t) \\ \dot{v}_x(t) \\ \dot{a}_x(t) \\ \dot{s}_y(t) \\ \dot{v}_y(t) \\ \dot{a}_y(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \begin{bmatrix} s_x(t) \\ v_x(t) \\ a_x(t) \\ s_y(t) \\ v_y(t) \\ a_y(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{B} \begin{bmatrix} j_x(t) \\ j_y(t) \end{bmatrix}. \tag{2}$$

This formulation gives time-stamped x/y-axis motion commands with the following state vector:

$$\boldsymbol{x}(t) = \left[s_x(t), v_x(t), a_x(t), s_y(t), v_y(t), a_y(t)\right]^T, \tag{3}$$

where $s_x(t)$ and $s_y(t)$ are the x-axis and y-axis displacements at time t, $v_x(t)$ and $v_y(t)$ are the axial velocities, and $a_x(t), a_y(t)$ are the axial accelerations. The input (control) vector is selected as the axis jerk profiles as

$$\boldsymbol{u}(t) = \left[j_x(t), j_y(t)\right]^T, \tag{4}$$

where $j_i(t)$ represents the ith axis instantaneous jerk.

The toolpath trajectory generation problem can then be formulated as a minimum time optimal control problem (OCP),

$$\min_{\boldsymbol{x},\boldsymbol{u},T} \int_0^T 1 \, \mathrm{dt} = \min T, \tag{5}$$

$$\left.\begin{aligned} \text{subject to:}\quad & \dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \\ & \boldsymbol{x}(t) \in \mathcal{X} \\ & \boldsymbol{u}(t) \in \mathcal{U} \end{aligned}\right\} \text{for t} \in [0, T], \tag{6}$$

with the objective to generate a toolpath with minimum time T that interpolates on the sequence of TCP coordinates, that is both dynamically feasible (in terms of velocity and actuation constraints), described by the sets $\mathcal{X}$ and $\mathcal{U}$, and constrained in terms of positional accuracy (i.e., TCP tolerance) – also captured within the state constraint set $\mathcal{X}$. This formulation, however, is ill posed in its current form as the position constraints $\mathcal{X}$ vary along the toolpath (see Fig. 2); commanded feedrate may change, and geometric tolerances may change within the part program.

To address the time varying problem definition, the OCP can be recast into a global one-shot optimisation of a chain of sequential subproblems, with each subproblem having its own state constraint set $\mathcal{X}_i$. The modified optimisation problem can be written as

$$\min_{\boldsymbol{x},\boldsymbol{u},T_i} \sum_{i=1}^{n_p} \left(T_i + \int_0^{T_i} \boldsymbol{x}^T(t) Q \boldsymbol{x}(t) + \boldsymbol{u}^T(t) R \boldsymbol{u}(t) dt\right), \tag{7}$$

$$\left.\begin{aligned} \text{subject to:}\quad & \dot{\boldsymbol{x}}(t) = f_i\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big) \\ & \boldsymbol{x}(t) \in \mathcal{X}_i \\ & \underbrace{v_x^2(t) + v_y^2(t)}_{\text{tooltip speed}} \le v_{\text{nom},i}^2 \\ & \underbrace{\begin{bmatrix} -a_{x,\max} \\ -a_{y,\max} \end{bmatrix} \le \begin{bmatrix} a_x(t) \\ a_y(t) \end{bmatrix} \le \begin{bmatrix} a_{x,\max} \\ a_{y,\max} \end{bmatrix}}_{\boldsymbol{x}(t) \in \mathcal{X}} \\ & \underbrace{\begin{bmatrix} -j_{x,\max} \\ -j_{y,\max} \end{bmatrix} \le \begin{bmatrix} j_x(t) \\ j_y(t) \end{bmatrix} \le \begin{bmatrix} j_{x,\max} \\ j_{y,\max} \end{bmatrix}}_{\boldsymbol{u}(t) \in \mathcal{U}} \end{aligned}\right\} \text{for } t \in [0, T_i] \tag{8}$$

where n_p represents the number of CL lines in the toolpath, T_i representing the time taken to traverse the ith CL line, Q and R being square penalty matrices used to penalise large state and control values, and with $\mathcal{X}_i$ and $v_{\text{nom},i}$ representing the position constraints and commanded feedrate along the ith cutter line, respectively. A global state constraint set $\mathcal{X}$ and input constraint set $\mathcal{U}$ has also been introduced to capture time-invariant constraints such as axial acceleration and axial jerk limits.

The function f_i denotes the zero-order hold (ZOH)-discretised system dynamics for the ith CL line,

$$\begin{aligned} \boldsymbol{x}(k+1) &= f_i(\boldsymbol{x}(k), \boldsymbol{u}(k), N_i) \\ &= (I + \Delta t_i A)\boldsymbol{x}(k) + \Delta t_i B \boldsymbol{u}(k), \text{ with } k = 1, \dots, N_i \end{aligned} \tag{9}$$

where I represents the identity matrix, and $\Delta t_i = T_i / N_i$ is the discretisation period for the ith CL line. Each segment is thus discretised into N_i equal time intervals between $[0, T_i]$, where T_i represents the final time for the ith segment.

Introducing the quadratic penalty matrices Q and R into the cost function enables optimisation of secondary objectives during toolpath generation such as minimising axial acceleration and jerk, whilst also promoting uniqueness of solutions in the optimisation problem. The expanded form of (6) becomes

$$\begin{aligned} &\boldsymbol{x}^T(t) Q \boldsymbol{x}(t) + \boldsymbol{u}^T(t) R \boldsymbol{u}(t) \\ &= \boldsymbol{x}^T(t) \mathrm{diag}([0, 0, \alpha, 0, 0, \alpha]) \boldsymbol{x}(t) + \boldsymbol{u}^T(t) \mathrm{diag}([\beta, \beta]) \boldsymbol{u}(t) \\ &= \alpha \left(a_x^2(t) + a_y^2(t)\right) + \beta \left(j_x^2(t) + j_y^2(t)\right) \quad \text{with } 0 < \alpha, \beta \le 1, \end{aligned} \tag{10}$$

where diag represents the diagonal matrix with the specified terms on the leading diagonal. With solely a minimum time toolpath optimisation objective, Q and R should be tuned to be small but nonzero values. With this addition, one guarantees that, in the case of multiple viable equal time

toolpaths, the optimal trajectory should be chosen as the trajectory with the least relative axial acceleration and jerk.

This formulation assumes that feed drive actuation limits remain constant, however one could also build the $\mathcal{X}$ and $\mathcal{U}$ constraint sets in a piecewise manner.

A direct multiple shooting approach [Kelley 2020] has been taken to solve the one-shot optimisation problem of (7), in which the control input (axial jerk) is kept constant between each $[k-1, k]$ time interval. The discretisation N_i may be chosen through geometric toolpath information, such as

$$N_i = \max\left\{\text{ceil}\left(\frac{L_i + 0.5F_iT_f}{10F_i\Delta t_s}\right), N_{\min}\right\}, \quad (11)$$

where L_i is the segment length (m), F_i is the commanded feedrate (m/s), and Δt_s is the NC interpolator sample period duration, in seconds. The term T_f represents the jerk limiting FIR filter delay time [Ward 2021] (in seconds), which is utilised as a geometry informed guess of the expected segment duration, including the cornering time.

Here, $N_{\min}$ is defined as the minimum number of discrete time intervals allocated to a segment and has been introduced to allow optimisation of short-segmented highly discretised toolpaths, in which the discretised segment duration $\text{ceil}\left(\frac{L}{F}\cdot\frac{1}{\Delta t_s}\right)$, measured in number of sample periods, would otherwise approach 1 sample, thereby causing undesirable singular control.

3.1 Formulation of TCP error constraints

The goal of trajectory generation for high-speed machining (HSM) is to generate the fastest TCP motion within specified geometric and kinematic tolerances. Therefore, without well-defined position tolerances, the motion will be constrained to simple point-to-point motion. This section will introduce the formulation of the TCP error constraints used within the global one-shot optimisation.

Prior research has constrained TCP position through the nonconvex cross product position constraint [Lin 2018],

$$\left\|\left(\vec{P}(t) - \vec{p}_i\right) \times \vec{d}_i\right\| \leq \varepsilon_{\text{Tol}} \text{ for } t \in [0, T_i], \quad (12)$$

where $\vec{P}(t) = \left(s_x(t), s_y(t)\right)^T$ is the instantaneous tooltip position, $\vec{p}_i$ is the initial CL point for the *i*th segment, and $\vec{d}_i$ is the unit-length direction vector between the $\vec{p}_i$ and $\vec{p}_{i+1}$ CL points. However, this constraint is nonconvex and therefore cannot be solved with guarantees on optimality (without exhaustive state space searches). To combat this, we propose a convex relaxation of the path tolerance through utilising a linear polytope constraint [Kouvaritakis 2016] as depicted in Fig. 2. In planar 2D toolpaths, this results in a bounding box composed of a set of linear inequalities centred about the nominal cutter line constraining the admissible toolpath trajectory region. The bounding box is configured per CL line and forms the position constraints within the state constraint set $\mathcal{X}_i$ for each *i*th segment. This approach can be extended to 3-axis toolpaths with an n-dimensional polytope where, as $n \to \infty$, the polytope constraint will become a tube of radius ε_{Tol} around the nominal cutter line.

One may observe in Fig. 2 that there exists an overlap region between the bounding boxes, denoted by the radius centred at p_2. This facilitates continuous toolpath motion during the transition between segments. In the case of short-segmented toolpaths, such overlap opens the possibility for global toolpath smoothing.

3.2 Scaling the optimisation problem

The scaling matrices Q and R also allow normalisation of secondary optimisation objectives (i.e., minimising axial acceleration and/or jerk) relative to the primary minimum time objective. This can be achieved through utilising a 'best guess' of the toolpath cycle time. As the total cycle time $T = \sum_i^{n_p} T_i$ is not known a priori, the normalisation can be accomplished through using the ideal cycle time, calculated using the assumption that the toolpath is traversed at its commanded feedrate as follows,

$$\alpha_i = \frac{J_{\text{accel}}}{N_i {a_{\max}}^2} \underbrace{\frac{1}{\left(\sum_i^{n_p} T_{v,i}\right)}}_{\text{ideal cycle time}}, \quad \beta_i = \frac{J_{\text{jerk}}}{N_i {j_{\max}}^2} \frac{1}{\left(\sum_i^{n_p} T_{v,i}\right)}, \quad (13)$$

with $J_{\text{accel}}, J_{\text{jerk}} \in (0,1]$ controlling the weights of the secondary performance objectives depicted in Equation (10).

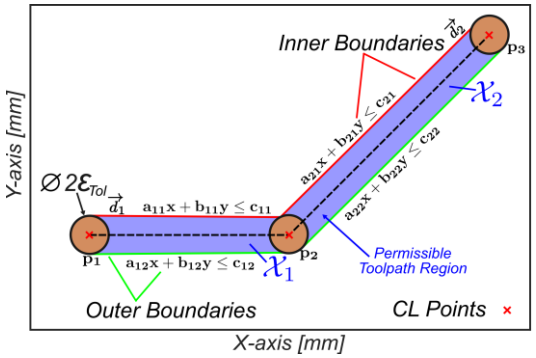

Fig. 2: Toolpath segments with piecewise state constraints representing geometric straightness and TCP tolerance.

The weights of the secondary objectives can be influenced through information provided within the STEP-NC part definition file, as depicted in Fig. 1. The problem in (7) has been implemented using CasADi [Andersson 2019] through direct transcription (using multiple shooting), transforming the one-shot optimisation a nonlinear programming (NLP) problem, using the CasADi Opti framework for NLP. The interior-point solver IPOPT [Wächter 2006] was used as the optimisation solver backend.

4 CASE STUDY: INDUSTRIAL TOOLPATH

This section presents performance validation of the proposed trajectory generation method against a commercially available CNC interpolator. The machine tool used for the benchmarking trials is the Hermle C52 machining centre [Maschinenfabrik Berthold Hermle AG 2024], equipped with a Siemens 840D SL numerical controller [Siemens AG 2019]. The interpolator parameters for the 840D were extracted from the system archive file, with an excerpt of the parameters visible in Tab. 1. Additionally, the interpolator has a sampling frequency of 500 hertz.

Tab. 1: 840D NC Interpolator Parameters

840D Surface Finish Setting	Maximum X-axis Acceleration [ms^{-2}]	Maximum Y-axis Acceleration [ms^{-2}]	Maximum X-axis Jerk [ms^{-3}]	Maximum Y-axis Jerk [ms^{-3}]
DYNROUGH	6	6	100	95
DYNFINISH	3	3	22	22

The trials on the Hermle C52 were performed with the 840D interpolator configured for DYNROUGH (roughing), and DYNFINISH (finishing). This interpolator setting (configured through the CYCLE832 cycle definition) modifies the system acceleration and jerk limits. As the benchmark toolpaths were planar (with x and y-axis motion only), only the relevant x/y-axis actuation limits have been listed. For ease of implementation, the minimum axial jerk limit was used such that the x-axis jerk was also limited to 95 ms^{-3}.

4.1 Case Study A: Minimum time formulation

This trial benchmarked the efficacy of the proposed trajectory generation algorithm in reducing machine cycle time. The on-machine trials have been performed on the Hermle C52, with the interpolator settings of Tab. 1. The proposed algorithm was used to generate toolpaths under the same constraints as depicted in Tab. 1. The generated paths are referred to as the 'optimised' paths from hereon.

As evidenced within Tab. 2, the proposed algorithm successfully achieves cycle time reduction across all trials, with reductions of up to 14.37% (trial A5) compared to the 840D interpolated paths. In addition, the constraint formulation of Section 3.1 has proven to be effective, with all trials being within 0.1% of the specified TCP tolerance.

The path RMSE was calculated through discretising the nominal toolpath (based on the toolpath CL points) with a minimum displacement of $\Delta s_{\min} = 0.1\varepsilon_{\mathrm{Tol}}$, after which the minimum Euclidean distance between the discretised path and the interpolated toolpaths was taken. The resulting path RMSE metric was then computed through

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\varepsilon_{\mathrm{interp},i})^2}{n}}, \tag{14}$$

where n is the number of points in the interpolated path, and $\varepsilon_{\mathrm{interp},i}$ is the ith path error, computed using pdist2 [The MathWorks, Inc. 2023]. Compared to the 840D interpolated path, the optimised path results in higher path RMSE across all trials, albeit staying within the prescribed TCP tolerance. The increase in toolpath performance comes at the expense of closeness to the nominal path, along with higher actuation. This is illustrated in Fig. 3a, which shows the difference in interpolated path geometry, and in Fig. 3c, which depicts the axial acceleration and jerk for the two paths. The industrial controller exceeds the recommended axial jerk limits. This may be due to kinematic optimisation functionality, which cannot be isolated nor deactivated. Further kinematic performance improvements are to be expected if one was to slacken the axial acceleration and jerk constraints in the generation of the optimised toolpaths.

4.2 Case Study B: Secondary objective modification

The second case study has been designed to demonstrate the efficacy of the formulation in (7), in terms of optimising for secondary objectives. Figure 4a depicts the trochoidal toolpath used for trials B1 to B4. A baseline trial was first completed on the 840D, along with a baseline minimum time objective trial - like Case Study A. The baseline trials were then compared to a minimum acceleration trial, and a minimum jerk trial, with the former resembling a minimum energy performance criterion, and the latter resembling a minimum curvature objective. Trial B1 was conducted on the Hermle C52 with the machine parameters listed in Table 1. The baseline minimum time trial (B2) has been computed with a relative weighting of $J_{\mathrm{accel}} = J_{\mathrm{jerk}} = 10^{-3}$, being a small nonzero positive value to solely promote uniqueness of solution.

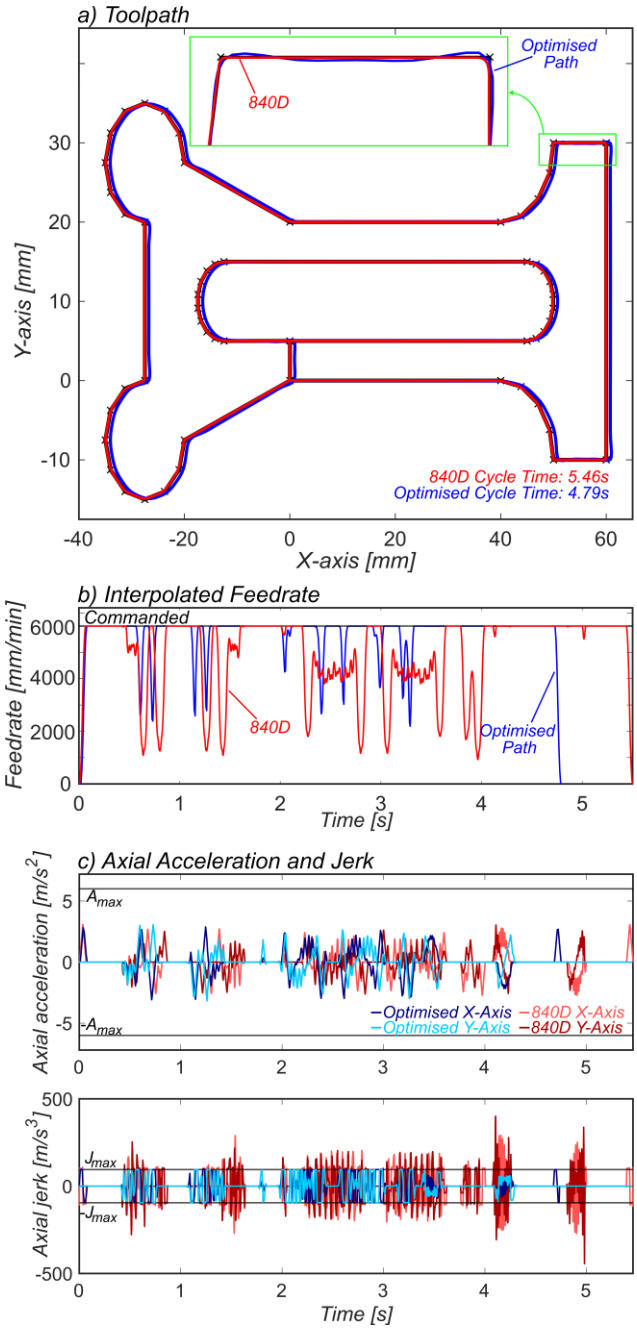

Fig. 3: Case Study A – Trial A4 Results

Tab. 2: Results for Case Study A – time-optimal objective.

#	Finish Type	Feedrate [mm/min]	TCP Tolerance [μm]	840D Cycle Time [s]	Optimised Cycle Time [s]	Cycle Time Reduction [%]	840D Path RMSE [μm]	Optimised Path RMSE [μm]	Optimised Toolpath Max. Error [μm]
A1	DYNROUGH	3000	50	9.44	9.22	2.33	16.80	17.31	49.96
A2	DYNROUGH	3000	100	9.39	9.15	2.56	16.81	46.80	99.91
A3	DYNROUGH	6000	50	5.45	5.06	7.16	18.71	28.05	50.03
A4	DYNROUGH	6000	100	5.46	4.79	12.27	17.80	53.55	99.97
A5	DYNFINISH	3000	10	13.36	11.44	14.37	1.33	5.45	10.00
A6	DYNFINISH	3000	50	10.36	9.59	7.43	17.09	24.53	50.02

Tab. 3: Results for Case Study B (cost function modification) with TCP tolerance 100 μm and feedrate 6000 mm/min.

#	Trial	Accel. Obj. [rel. units]	Jerk Obj. [rel. units]	Cycle Time [s]	Path RMSE [μm]	Path RMS Accel. [ms^{-2}]	Path RMS Jerk [ms^{-3}]
B1	840D DYNROUGH	-	-	13.36	24.61	1.16	79.23
B2	Optimised; Min. Time	10^{-3}	10^{-3}	11.62	41.80	1.49	52.05
B3	Optimised; Min. Acceleration	10^{6}	10^{-3}	11.79	39.75	1.49	52.80
B4	Optimised; Min. Jerk	10^{-3}	10^{6}	11.73	39.63	1.54	55.20

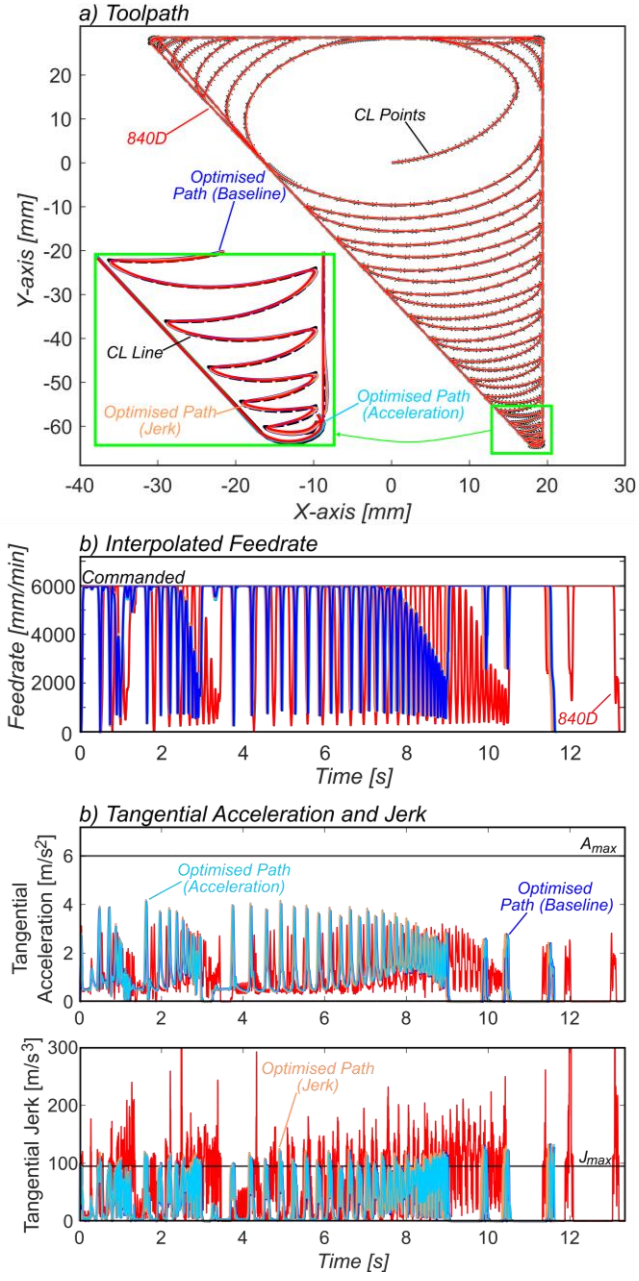

Fig. 4: Case Study B Results.

An arbitrary large weighting of $J = 10^6$ was used for trials B3 and B4 to demonstrate the efficacy of including the secondary optimisation objectives.

As displayed in Tab. 3, the proposed algorithm outperformed the industrial controller across all trials in terms of cycle time. The improved performance is reflected through path RMS acceleration and jerk metrics, with the optimised path warranting greater actuation (axial acceleration) to assist in reducing cycle time. The resultant (tangential) jerk exceeded the specified jerk limit. However, the individual axial jerk signals remained within axial limits, thereby still conforming to specified kinematic constraints.

5 CONCLUSION AND FUTURE WORK

In this work, we have introduced a digital thread enabled optimal trajectory generation framework that allows multi-objective optimisation of minimum time toolpaths.
The algorithm has been benchmarked against a commercial numerical controller, achieving cycle time reductions of up to 14.37%. The proposed trajectory optimisation algorithm is flexible and allows optimisation of secondary performance objectives that are a linear combination of the system states. Further work is required to incorporate frequency domain objectives into the optimisation problem, and to intelligently tune the multi-objective optimisation.

In addition, we have demonstrated a framework that exploits the STEP-NC standard to customise the trajectory generation algorithm prior to machining. Through modifying the STEP-NC part program file, one may supply the optimisation-based interpolator with qualitative performance objectives, such as prioritising straightness in the minimum time toolpath generation. The framework allows constraining of TCP tolerance and axial motion derivatives (up to jerk) with a different constraint on each individual CL line. Whilst the presented one-shot optimisation can be shown to be near time optimal, the computation of a global one-shot toolpath optimisation is costly. A suitable application of the algorithm is in the context of high-volume manufacturing, where the cycle time reduction of an optimisation-based trajectory generation algorithm heavily outweighs the costly one-off computation. Further research is required into reducing the computational complexity of the problem through intelligent windowed optimisation, or parallel computation of the toolpath. Subsequent research should aim to provide a proof for the optimality of the formulation.

6 ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) [studentship project reference 2884421].

7 REFERENCES

[Abdel-Aty 2024] Abdel-Aty, T. A., Negri, E.: 'Conceptualizing the digital thread for smart manufacturing: a systematic literature review'; Journal of Intelligent

Manufacturing, 35, 8 (2024), 3629–3653. https://doi.org/10.1007/s10845-024-02407-1

[Altintas 2012] Altintas, Y.: 'Manufacturing automation: Metal cutting mechanics, machine tool vibrations, and CNC design'; (2nd ed.). Cambridge: Cambridge University Press (2012).

[Andersson 2019] Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., Diehl, M.: 'CasADi: a software framework for nonlinear optimization and optimal control'; Mathematical Programming Computation, 11, 1 (2019), 1–36. https://doi.org/10.1007/s12532-018-0139-4

[Bosetti 2016] Bosetti, P., Ragni, M.: 'Milling Part Program Preprocessing for Jerk-limited, Minimum-time Tool Paths Based on Optimal Control Theory'; IEEJ Journal of Industry Applications, 5, 2 (2016), 53–60. https://doi.org/10.1541/ieejjia.5.53

[Lin 2018] Lin J., Somani, N., Hu, B., Rickert, M., and Knoll, A.: 'An Efficient and Time-Optimal Trajectory Generation Approach for Waypoints Under Kinematic Constraints and Error Bounds'. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5869–5876. https://doi.org/10.1109/IROS.2018.8593577

[Cheng 2022] Cheng, K., Zhao, G., Wang, W., Liu, Y.: 'An estimation methodology of energy consumption for the intelligent CNC machining using STEP-NC'; The International Journal of Advanced Manufacturing Technology, 123, 1 (2022), 627–644. https://doi.org/10.1007/s00170-022-10194-3

[Erkorkmaz 2001] Erkorkmaz, K., Altintas, Y.: 'High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation'; International Journal of Machine Tools and Manufacture, 41, 9 (2001), 1323–1345. https://doi.org/10.1016/S0890-6955(01)00002-5

[Hu 2023] Hu, Y., Jiang, X., Huo, G., Su, C., Zhou, S., Wang, B., et al.: 'A novel feed rate scheduling method with acc-jerk-continuity and round-off error elimination for non-uniform rational B-spline interpolation'; Journal of Computational Design and Engineering, 10, 1 (2023), 294–317. https://doi.org/10.1093/jcde/qwad004

[Imad 2022] Imad, M., Hopkins ,C., Hosseini ,A., Yussefian ,N.Z., and Kishawy, H. A.: 'Intelligent machining: a review of trends, achievements and current progress'; International Journal of Computer Integrated Manufacturing, 35, 4–5 (2022), 359–387. https://doi.org/10.1080/0951192X.2021.1891573

[International Organization for Standardization 2009] International Organization for Standardization: 'ISO 6983-1:2009'; (2009). Retrieved 19 April 2025, from https://www.iso.org/standard/34608.html

[[International Organization for Standardization 2019] [International Organization for Standardization: 'ISO 10303-235:2019'; (2019). Retrieved 18 April 2025, from https://www.iso.org/standard/74409.html

[International Organization for Standardization 2021] International Organization for Standardization: 'ISO 10303-42:2021(E)'; (2021). Retrieved 19 April 2025, from https://www.steptools.com/stds/smrl/data/resource_docs/geometric_and_topological_representation/sys/4_schema.htm#geometry_schema.bounded_curve

[International Organization for Standardization 2022] International Organization for Standardization: 'ISO 10303-238:2022'; (2022). Retrieved 19 April 2025, from https://www.iso.org/standard/84898.html

[Kelley 2020] Kelley, M. T., Baldick, R., Baldea, M.: 'A direct transcription-based multiple shooting formulation for dynamic optimization'; Computers & Chemical Engineering, 140 (2020), 106846. https://doi.org/10.1016/j.compchemeng.2020.106846

[Kim 2024] Kim, H., Kontar, R. A., Okwudire, C. E.: 'Intelligent Feedrate Optimization Using an Uncertainty-Aware Digital Twin Within a Model Predictive Control Framework'; IEEE Access, 12 (2024), 49947–49961. https://doi.org/10.1109/ACCESS.2024.3384471

[Kim 2020] Kim, H., Okwudire, C. E.: 'Simultaneous servo error pre-compensation and feedrate optimization with tolerance constraints using linear programming'; The International Journal of Advanced Manufacturing Technology, 109, 3 (2020), 809–821. https://doi.org/10.1007/s00170-020-05651-w

[Kouvaritakis 2016] Kouvaritakis, B.: 'Model predictive control: classical, robust and stochastic'; Cham: Springer (2016).

[Liu 2018] Liu, Y., Zhao, G., Zavalnyi, O., Cao, X., Cheng, K., Xiao, W.: 'STEP-Compliant CAD/CNC System for Feature-Oriented Machining'; Computer-Aided Design and Applications, 16, 2 (2018), 358–368. https://doi.org/10.14733/cadaps.2019.358-368

[Maschinenfabrik Berthold Hermle AG 2024] Maschinenfabrik Berthold Hermle AG: 'Hermle C52 Machining Centre'; (n.d.). Retrieved 15 July 2024, from https://www.hermle.de/en/machining-centres-automation/models/machining-centre-c-52/

[Nguyen 2009] Nguyen, V. K., Stark, J.: 'STEP-compliant CNC Systems, Present and Future Directions'; In X. Xu & A. Y. C. Nee (Eds.), Advanced Design and Manufacturing Based on STEP. London: Springer (2009), 215–231. https://doi.org/10.1007/978-1-84882-739-4_10

[Raoufi 2024] Raoufi, K., Sutherland, J. W., Zhao, F., Clarens, A. F., Rickli, J. L., Fan, Z., et al.: 'Current state and emerging trends in advanced manufacturing: smart systems'; The International Journal of Advanced Manufacturing Technology, 134, 7 (2024), 3031–3050. https://doi.org/10.1007/s00170-024-14279-z

[Rauch 2012] Rauch, M., Laguionie, R., Hascoet, J.-Y., Suh, S.-H.: 'An advanced STEP-NC controller for intelligent machining processes'; Robotics and Computer-Integrated Manufacturing, 28, 3 (2012), 375–384. https://doi.org/10.1016/j.rcim.2011.11.001

[Siemens AG 2019] Siemens AG: 'SINUMERIK 840D sl: CNC Commissioning—NC, PLC, Drive'; Siemens (2019). Retrieved from https://cache.industry.siemens.com/dl/files/178/109769178/att_991299/v1/840Dsl_CNC_commiss_man_en-US.pdf

[Suh 2003] Suh, S. H., Lee, B. E., Chung, D. H., Cheon, S. U.: 'Architecture and implementation of a shop-floor programming system for STEP-compliant CNC'; Computer-Aided Design, 35, 12 (2003), 1069–1083. https://doi.org/10.1016/S0010-4485(02)00179-3

[Tajima 2022] Tajima, S., Sencer, B.: 'Online interpolation of 5-axis machining toolpaths with global blending'; International Journal of Machine Tools and Manufacture, 175 (2022), 103862. https://doi.org/10.1016/j.ijmachtools.2022.103862

[Tajima 2018] Tajima, S., Sencer, B., Shamoto, E.: 'Accurate interpolation of machining tool-paths based on FIR filtering'; Precision Engineering, 52 (2018), 332–344. https://doi.org/10.1016/j.precisioneng.2018.01.016

[The MathWorks, Inc., 2023] The MathWorks, Inc.: 'pdist2'; (2023). Retrieved 22 April 2025, from https://uk.mathworks.com/help/stats/pdist2.html

[Wächter 2006] Wächter, A., Biegler, L. T.: 'On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming'; Mathematical Programming, 106, 1 (2006), 25–57. https://doi.org/10.1007/s10107-004-0559-y

[Wang 2025] Wang, Y., Hu, C., Li, Z., He, Z., Lin, S., Wang, Y., et al.: 'On the consistency of path smoothing and trajectory planning in CNC machining: A surface-centric evaluation'; Robotics and Computer-Integrated Manufacturing, 92 (2025), 102873. https://doi.org/10.1016/j.rcim.2024.102873

[Ward 2021] Ward, R., Sencer, B., Jones, B., Ozturk, E.: 'Accurate prediction of machining feedrate and cycle times considering interpolator dynamics'; The International Journal of Advanced Manufacturing Technology, 116, 1 (2021), 417–438. https://doi.org/10.1007/s00170-021-07211-2

[Ward 2022] Ward, R., Sencer, B., Panoutsos, G., Ozturk, E.: 'On-The-Fly CNC interpolation using frequency-domain FFT-based filtering'; Procedia CIRP, 107 (2022), 1571–1576. https://doi.org/10.1016/j.procir.2022.05.193

[Yan 2023a] Yan, G., Zhang, D., Xu, J., Sun, Y.: 'A C3 continuous double circumscribed corner rounding method for five-axis linear tool path with improved kinematics performance'; Journal of Computational Design and Engineering, 10, 4 (2023a), 1490–1506. https://doi.org/10.1093/jcde/qwad066

[Yan 2023b] Yan G., Zhang D., Xu J., Sun Y.: 'Corner smoothing for CNC machining of linear tool path: A review'; Journal of Advanced Manufacturing Science and Technology, 3, 2 (2023b), 2023001–2023001. https://doi.org/10.51393/j.jamst.2023001

[Yuan 2022] Yuan, C., Li, G., Kamarthi, S., Jin, X., Moghaddam, M.: 'Trends in intelligent manufacturing research: a keyword co-occurrence network based review'; Journal of Intelligent Manufacturing, 33, 2 (2022), 425–439. https://doi.org/10.1007/s10845-021-01885-x

[Zhang 2019] Zhang, J., Ding, J., Li, Q., Ding, Q., Jiang, Z., Wang, W., Du, L.: 'An Optimization Model of High-Speed and High-Precision Machining Based on Model Predictive Control'; Presented at the ASME 2018 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers Digital Collection (2019). https://doi.org/10.1115/IMECE2018-88326