Smooth Trajectory Generation for 5-Axis CNC Machine Tools

by Rob Ward

Dept of Automatic Control and Systems Engineering

Industrial Doctorate Centre in Machining Science

Advanced Manufacturing Research Centre

University of Sheffield

07 January 2022

A thesis submitted for the degree of Doctor of Engineering.

Abstract

This thesis is presented in the alternative thesis format. The first paper presents an accurate machining feedrate prediction technique by modeling the trajectory generation behaviour of modern CNC machine tools. Typically, CAM systems simulate machines' motion based on the commanded feedrate and the path geometry. Such approach does not consider the feed planning and interpolation strategy of the machine's numerical control (NC) system. In this study, trajectory generation behaviour of the NC system is modelled and accurate cycle time prediction for complex machining toolpaths is realized. NC system's linear interpolation dynamics and commanded axis kinematic profiles are predicted by using Finite Impulse Response (FIR) based low-pass filters. The corner blending behaviour during non-stop interpolation of linear segments is modeled, and for the first time, the minimum cornering feedrate, that satisfies both the tolerance and machining constraints, has been calculated analytically for 3-axis toolpaths of any geometry. The proposed method is applied to 4 different case studies including complex machining tool-paths. Experimental validations show actual cycle times can be estimated with >90% accuracy, greatly outperforming CAM-based predictions. It is expected that the proposed approach will help improve the accuracy of virtual machining models and support businesses decision making when costing machining processes.

The second paper presents a novel real-time interpolation technique for 5-axis machine tools to attain higher speed and accuracy. To realize computationally efficient real-time interpolation of 6DOF tool motion, a joint workpiece-machine coordinate system interpolation scheme is proposed. Cartesian motion of the tool centre point (TCP) is interpolated in the workpiece coordinate system (WCS), whereas tool orientation is interpolated in the machine coordinate system (MCS) based on the FIR filtering. Such approach provides several advantages: i) it eliminates the need for complex real-time spherical interpolation techniques, ii) facilitates efficient use of slower rotary drive kinematics to compensate for the dynamic mismatch between Cartesian and rotary axes and achieve higher tool acceleration, iii) mitigates feed fluctuations

while interpolating near kinematic singularities. To take advantage of such benefits and realize accurate joint WCS-MCS interpolation scheme, tool orientation interpolation errors are analysed. A novel approach is developed to adaptively discretize long linear tool moves and confine interpolation errors within user set tolerances. Synchronization errors between TCP and tool orientation are also characterized, and peak synchronization error level is determined to guide the interpolation parameter selection. Finally, blending errors during non-stop continuous interpolation of linear toolpaths are modelled and confined. Advantages of the proposed interpolation scheme are demonstrated through simulation studies and validated experimentally. Overall, proposed technique can improve cycle times up to 10% while providing smooth and accurate non-stop real-time interpolation of tool motion in 5-axis machining.

The third paper proposes a novel online interpolation method for 3 and 5-axis machine tools to reduce machining cycle times. Previous Finite Impulse Response filtering based methods for numerically controlled machining used the maximum feedrate command within a part program for selecting the FIR filter time constant resulting in sub-optimal kinematic performance for toolpaths with varying feedrates. This paper presents an On-The-Fly (OTF) method of NC interpolation capable of kinematically optimising each individual G01 command. The method adaptively changes the FIR filter time constant along the toolpath maximising the kinematic performance for each G01 command without violating the constraints thereby reducing the overall machining cycle time. The tool centre point and orientation blending errors during continuous machining are controlled using an Overlap-Add (OLA) method of signal reconstruction. The OLA method is analytically calculated to confine interpolation errors within user set tolerances. The reduction in machining cycle times compared to standard FIR based interpolation methods is demonstrated through simulation studies. The proposed OTF method of NC interpolation can reduce continuous and P2P machining cycle times by up to 5% and 7% respectively while generating accurate online adaptively interpolated 3 and 5-axis reference trajectories.

Finally, Finite Impulse Response filtering is increasingly becoming the interpolation method of choice in modern computer numerically controlled (CNC) machining centres. The method offers significant computational advantages over polynomial based methods. Most published methods use fixed FIR filter time constants to smooth the input signal. Recently, On-The-Fly interpolation was presented using Direct Convolution methods to adaptively change and optimise the FIR filter time constant throughout the toolpath online. Direct convolution in

the time domain is an efficient method of implementing FIR interpolation online, however, computational advantages can be gained by using frequency domain methods instead. This research introduces a novel on-the-fly CNC interpolation method using Fast Fourier Transforms (FFTs). The presented OTF FFT method demonstrates an order increase in computational speed than the direct convolution OTF method. The effectiveness of the proposed method is validated in simulation based case studies.

Acknowledgements

I would like to thank my research supervision team, Dr Burak Sencer, Dr Bryn Jones and Dr Erdem Ozturk. In particular, I would like to express my sincere gratitude to Dr Burak Sencer for his valuable guidance, support and patience which he has provided throughout the past couple of years. I look forward to collaborating together for years to come. To Dr Erdem Ozturk for providing industrial support and collaboration opportunities and to Dr Bryn Jones for his academic guidance and attention to detail.

I wish to thank my colleagues in the Advanced Manufacturing Research Centre and Industrial Doctorate Centre for sharing their knowledge and experience with me. The centre has provided an abundance of opportunities during the period of the EngD. From the initial integration within the Machining Dynamics research team through to a position in the Digital Machining team I have been exposed to the frontline of industry's manufacturing challenges. I would like to thank the following, Dr David Curtis, Omer Ozkirimli, Dr Chao Sun, Dr Javier Dominguez-Caballero, Dr Pete Crawforth, Dr Tom Mcleay, Dr Sabino Ayvar-Soberanis, Phil Bell, Dr Jon Stammers, Clare Clarke, Jodie Greaves, Emily Pickford, Adam Brown and Rich Bonnell.

Finally, thank you to my partner, family and friends for supporting me throughout.

Contents

Chapter 1

Introduction

1.1 Motivation

The adoption of Industry 4.0 technologies in the manufacturing sector is vital if companies are to remain competitive in the future green net-zero marketplace. Many innovative technologies are being implemented such as Digital Twins, Cyber-Physical and Edge-based systems, Cloud Computing and many more. The key enabler has been the rollout of monitoring and communication systems to the shopfloor. Data is now available in abundance, however, the potential usefulness of this data still remains sub-optimal. One technology attempting to encapsulate the smart manufacturing data is the Digital Thread. The term refers to the data and communication framework which connects the physical and digital information of a product across the product's life cycle. The objectives are many, but traceability is the fundamental purpose as it enables the ability to highlight or predict areas of improvement or non-conformance with minimal interventions such as testing.

In terms of CNC machining, the digital thread that surrounds the CAD-CAM-CNC cycle is shown in Fig.1.1. Over the next few years significant research will be conducted in this domain such that through modelling, measurement and prediction, it is feasible to predict final part quality and conformance in terms of geometric accuracy, surface finish and part distortion.

In order to demonstrate where the main body of this thesis fits into the larger digital machining ecosystem, an overview of the machining process is presented. The generic machining process is shown in Fig.1.1 where it presents the discrete stages in the production cycle. First, the part to be manufactured is modelled in a computer aided design (CAD) software package.

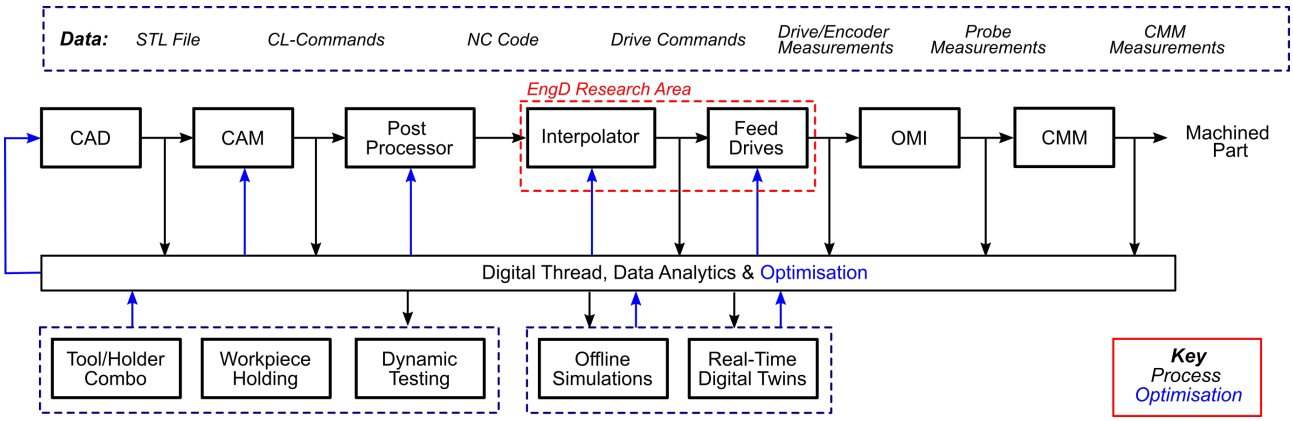

Figure 1.1: Stages in the Machining Process

The toolpath is generated by computer aided manufacturing (CAM) software with user defined inputs such as toolpath strategy, tool specifications, feed per tooth, stepover and many more machining parameters. The generic machine-independent output is a series of cutter locations (CL), spindle speeds and feedrate commands. The CL commands are usually in the form of CL or Automatically Programmed Tool (APT) source files. The file is post processed to convert the commands into numerical control (NC) code specific for a machine tool and controller type. The NC code is usually verified collision safe offline through software such as Vericut prior to machining. The NC code is then loaded onto the machining centre and read/parsed by the interpolator which is housed in the NC unit (NCU). For a standard 5-axis machine tool, the interpolator generates motion control commands for the 5 machine feed drives (3 translational, 2 rotary) and velocity control commands for the spindle. The closed loop feed drive control systems move the machine axes in response to the nominal or reference commands generated by the interpolator. Many compensation strategies are applied by the control systems such as feedforward control for quadrant error and friction compensation. During machining the drive encoders and NC information can be read by monitoring systems and used for real-time Digital Twins and shopfloor data systems. Finally, on-machine inspection (OMI) or post machining inspection by coordinate measuring machine (CMM) may take place as part of the quality process to test the machined part against the design intent. To accurately predict the final part quality as is envisaged with the machining digital thread, it is vital to model and validate each stage in the machining process shown in Fig.1.1.

The research conducted during the EngD period encompassed two main areas of digital machining, namely NC interpolation and machining Digital Twins. The thesis presents the body of work on NC interpolation and the Digital Twin research is captured separately in the published

works.

To set the scene, originally the EngD research project was tasked to use data generated from the machining process to optimise future machining operations. In particular, Iterative Learning Control (ILC) [1][2] was to be implemented to optimise machining processes. Significant work was conducted which integrated machining centre monitoring systems with external sensors such as force dynamometers and real-time Digital Twins. The general principle of using ILC is to modify the input of a process using data from the previous input and the measured output. Over time the error between the desired output and actual output would tend to zero by using the process errors from previous operations to optimise the next input. The ILC method was demonstrated experimentally to feedrate scheduling using measured cutting forces and spindle loads. In particular, the process was applied to pocketing toolpaths as shown in Chapter 2. However, despite significant attempts to tune the controller, the method was unsuccessful. On further analysis, it was demonstrated there were significant differences between commanded and actual feedrates during machining trials. The results showed a linear one-to-one mapping did not exist between the NC code commands and the feed drive commands and the responses were highly dependent on tool centre point (TCP) tolerance and feedrate. Despite the significant effort and resources invested into the ILC machining trials, this proved to be a major turning point in the EngD research and resulted in a significant shift in focus. The research effort turned to the role of the interpolator in machine response. In industry, however, NC interpolators are black boxes with commercially protected and patented algorithms making it very challenging to fully understand the internal process. Therefore, it was vital to understand the factors that influence the achievable feedrate and ultimately machining cycle time. Starting with feedrate prediction via analytical modelling, the research developed new methods of NC interpolation for both 3 and 5 axis machining. The driver has always been industry focused with the main effort to reduce machining cycle times whilst maintaining part geometric accuracy.

The chapters in this thesis are self contained and comprehensive. The material was developed in the order as presented and it is recommended they are approached in this manner. I hope the reader enjoys the material as much as I had developing and implementing it.

1.2 Objectives

The aim of the study was to develop accurate and efficient real-time interpolation methods for CNC machine tools. The objectives were as follows:

1. Develop a method of predicting feedrate and machining cycle times for dynamic milling strategies considering tool centre point tolerance and machine kinematic limits.

2. Develop a method of accurate real-time interpolation for 3-axis machining considering tool centre point interpolation blending errors.

3. Develop a method of accurate real-time interpolation for 5-axis machining considering tool centre point position and tool orientation interpolation blending errors.

4. Propose a method to characterise and compensate for nonlinear interpolation errors during 5-axis machining.

5. Develop a method of computationally efficient real-time interpolation for future machine tool requirements.

1.3 Outline of thesis

The thesis has been submitted in the University of Sheffield "Alternative Format" thesis style. As such, it is written and structured with four sequential chapters which have been published, accepted or ready for publication at the time of submission, forming a coherent body of work.

The thesis is presented as follows:

- Chapter 2 presents published work titled *Accurate prediction of machining feedrate and cycle time considering interpolator dynamics*. The research developed a new method of 3-axis NC interpolation and applied this to both feedrate prediction and machining cycle time estimation.

- Chapter 3 presents published work *Five-axis trajectory generation considering synchronisation and nonlinear interpolation errors*. The paper extends the interpolation method to 5-axis and introduces a method of controlling the tool orientation blending errors. The research also characterises the nonlinear interpolation error generated from interpolating rotary motions in the machine coordinate system and presents a method to compensate.

- Chapter 4 presents work ready for submission titled *On-The-Fly Trajectory Generation for 3 and 5-Axis CNC Interpolation*. The paper introduces On-The-Fly interpolation which optimises the kinematic performance of each individual CL command throughout the toolpath. The proposed method is able to reduce machining cycle times whilst satisfying kinematic constraints and tolerance requirements for both 3 and 5-axis machining.

- Chapter 5 presents work accepted for conference titled *On-The-Fly CNC Interpolation using Frequency-Domain FFT-based Filtering*. The paper finalises the research by demonstrating an interpolation method to conduct On-The-Fly CNC interpolation in the frequency domain which significantly reduces the online computational cost.

- Chapter 6 presents the conclusions and recommendations for further work.

- Appendix A presents original work featuring the derivation of the forward and inverse kinematics for a non-orthogonal machine tool. The material supports the 5-axis research presented in chapter 4.

1.4 Chapter Contributions

A summary of the novel contributions and differences between each chapter is hereby presented.

1.4.1 Chapter 2

The original contributions to research are as follows:

1. A novel method of FIR filtering using matched FIR filters was developed. The linear interpolation dynamics and commanded axis kinematic profiles of NC systems were predicted using both 2 and 3 first order FIR filters with matching time constants. Previous works had used unmatched FIR filter time constants. The new method allowed the design process to be reduced to a single parameter to based on the kinematic constraint (machine tool jerk setting). The method removes the need for system testing which saves significant resources and prevents machine downtime during the production cycle.

2. The corner blending behaviour during non-stop interpolation of linear segments was modeled by introducing velocity blending pulses. The cornering time was linked to the velocity blending pulse to control the cornering velocity of the interpolated TCP feedrate.

3. Derived from the convolution integral, the analytical derivations of the interpolated kinematic profiles were used to determine velocity blending pulse properties. Significantly, TCP position blending error during cornering segments was modelled and indirectly provided a method to satisfy TCP tolerance through velocity blending pulse control.

4. For the first time, the minimum cornering feedrate, that satisfies both the tolerance and machining constraints, was calculated analytically for toolpaths of any geometry.

5. The FIR-based linear interpolation method was applied to feedrate prediction. The proposed method demonstrated cycle times can be estimated with >90% accuracy, greatly outperforming CAM-based predictions. The feedrate prediction method was validated experimentally against four different case studies demonstrating industrial 3-axis machining tool-paths.

6. The predicted feedrate method was incorporated into a cutting force model, demonstrating an increase in cutting force accuracy for a complex toolpath. The combined model was validated experimentally through machining trials.

1.4.2 Chapter 3

The original contributions to research are as follows:

1. A novel real-time interpolation technique for 5-axis machine tools was developed. The research extended the FIR filtering based method from chapter 2. The key difference is the extension to 5-axis from 3-axis machining.

2. A joint workpiece-machine coordinate system interpolation scheme based on FIR filtering was created. The method eliminated the requirement for complex real-time spherical interpolation techniques while mitigating feed fluctuations when interpolating near kinematic singularities.

3. The blending errors during non-stop continuous interpolation of linear toolpaths were modelled and confined using the rotary feed override factor. In particular, the MCS rotary axis blending error was linked to the WCS tool orientation blending error through a Jacobian linearisation based method. The rotary feed override factor was calculated using analytical derivations of the convolution integral thereby satisfying TCP and ORI user defined tolerances.

4. For the first time, the Cartesian and rotary FIR filter time constants were decoupled and designed separately. In doing so, the performance of machine tool feed drives were maximised. Significant reductions in machining cycle times (up to 10%) were demonstrated and validated through experimental trials.

5. Synchronisation between the tool orientation and TCP position was explicitly addressed. In particular, it was shown that synchronisation errors can be mitigated when decoupling the Cartesian and rotary FIR filter time constants through a designed alignment time delay. The heuristic selection of the optimal FIR time constant was presented based on the characterised synchronisation between the TCP and ORI motions.

6. A method of characterising nonlinear orientation interpolation errors caused by linearly interpolating rotary axis positions was presented. For the first time, maritime navigation methods were linked to CNC interpolation.

7. To compensate for the nonlinear tool orientation interpolation errors a new approach to adaptively discritising the ORI trajectory confining interpolation errors to within user defined tolerances was developed.

1.4.3 Chapter 4

The original contributions to research are as follows:

1. A method of real-time on-the-fly interpolation was developed for both 3 and 5-axis machining toolpaths. It introduced the use of an adaptive interpolation scheme based on FIR filtering.

2. Previous methods of FIR interpolation used the maximum feedrate and drive kinematic constraints to determine a worst case FIR filter time constant that would ensure the kinematic constraints are satisfied throughout the whole toolpath. A new developed method segments each line of G-code and kinematically optimises each commanded segment during toolpath motion. This leads to each feed drive maximising the kinematic performance and reduces overall machining cycle times.

3. The new on-the-fly method used direct convolution in the time domain and for the first time, used an overlap and add method of signal reconstruction to control the overlap in smoothed segmented velocity signals. The TCP and ORI blending errors were analytically modelled and confined using the overlap time of the reconstructed interpolated signals.

4. For the first time in FIR CNC interpolation research, the consecutive feedrates in a part program were assumed not equal. Previous methods assumed equal feedrates using the larger feedrate and therefore overestimated the TCP blending error. With the unequal feedrate assumption, a new method to calculate the TCP and ORI errors was determined analytically. Dwell between velocity pulses was used as a means to control TCP and ORI errors as opposed to blending velocity pulses from chapter 2 and 3. The new method showed more accurate TCP and ORI tolerances could be met leading to further reductions in cycle times compared to the over conservative existing dwell based methods.

5. Validation simulations on 3 and 5-axis machining toolpaths were presented. The results show a >5% and 7% reduction in machining cycle times compared to standard FIR interpolation in 5-axis continuous and P2P machining respectively. In 3-axis, approximately 4% and 7% reduction in cycle times was demonstrated during continuous and P2P machining respectively.

1.4.4 Chapter 5

The original contributions to research are as follows:

- For the first time, frequency domain methods of convolution was implemented to interpolate NC toolpaths - this is in contrast to previously published FIR time domain methods such as chapters 2 to 4 of this thesis.

- The block transform method of circular convolution using Fast Fourier Transforms was used to control the overlap between the segmented interpolated velocity signals. The overlap controlled the cornering velocity and TCP blending error.

- A comparison of interpolation methods was conducted in terms of computational efficiency. It was shown that FFT based methods of convolution demonstrate significantly faster computational times per interpolated output than direct convolution methods in the time domain.

- Finally, it was demonstrated for NC interpolation of toolpaths that FFT based methods can be further optimised to reduce time of computation by selecting input and filter lengths to match exponential to the base 2 whole numbers.

1.5 Additional Contributions

This section states the publications conducted and the co-author contributions during the EngD.

1.5.1 Journal Publications during the EngD

The following works were conducted during the EngD period Sept 17 - Dec 21.

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

2. **Ward, R.**, Sencer, B., Jones, B., Ozturk, E. Five-axis trajectory generation considering synchronisation and nonlinear interpolation errors. Journal of Manufacturing Science and Engineering, Transactions of the ASME (2022). https://doi.org/10.1115/1.4053460.

3. **Ward, R.** Sun, C., Dominguez-Caballero, J., Ojo, S., Ayvar-Soberanis, S., Curtis, D., Ozturk, E. Machining Digital Twin using real-time model-based simulations and lookahead function for closed loop machining control. International Journal of Advanced Manufacturing Technology (2021). https://doi.org/10.1007/s00170-021-07867-w

4. **Ward, R.**, Soulatiantork P, Finneran S, Hughes R, Tiwari A. Real-time vision-based multiple object tracking of a production process: Industrial digital twin case study. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture. 2021;235(11):1861-1872. https://doi.org/10.1177/09544054211002464

5. Sun, C., Dominguez-Caballero, J., **Ward, R.**, Ayvar-Soberanis, S., Curtis, D. (2021). Machining Cycle Time Prediction: Data-driven Modelling of Machine Tool Feedrate Behavior with Neural Networks. Journal of Robotics and Computer-Integrated Manufacturing. https://doi.org/10.1016/j.rcim.2021.102293.

1.5.2 Conference Proceedings

1. **Ward, R.**, Ozkirimli, O., Jones, B. Increasing Part Geometric Accuracy in High Speed Machining using Cascade Iterative Learning Control, Procedia CIRP. 101 (2021) 298–301. https://doi.org/10.1016/j.procir.2020.10.006.

2. **Ward, R.**, Sencer, B., Panoutsos, G., Ozturk, E. On-The-Fly CNC Interpolation using Frequency-Domain FFT-based Filtering, CIRP CMS 2022.

1.5.3 Candidate Statement

I confirm that the work submitted is my own, except where work that has formed part of jointly authored publications has been included. The contribution of the other authors to this work has been explicitly indicated below. I confirm that appropriate credit has been given within the thesis where reference has been made to the work of others.

1.5.4 Co-author Contributions

In all publications, I planned, and wrote the manuscripts whilst the supervisors as co-authors reviewed and provided comments on the manuscripts. The following members of staff provided supervision during the EngD research:

- Dr Burak Sencer - External supervisor (Mar 20 to Dec 21)
- Dr Bryn Jones - Academic supervisor (Sep 17 to Sep 21)
- Dr George Panoutsos - Academic supervisor (Oct 21 to Dec 21)
- Dr Erdem Ozturk - AMRC / industrial supervisor (Mar 20 to Dec 21)
- Omer Ozkirimli - AMRC / industrial supervisor (Apr 19 - Mar 20)
- Dr Tom Mcleay - AMRC/ industrial supervisor (Sep 17 - Apr 19)

Chapter 2

Journal Paper 1 - Accurate Prediction of Machining Feedrate and Cycle Time Considering Interpolator Dynamics

2.1 Introduction

With the introduction of concepts like virtual manufacturing [3] and digital twins [4], building process models and predicting actual machining process conditions in the computer environment has become paramount in attaining higher productivity and throughput in today's manufacturing. For example, accurate machining cycle time prediction is vital for industry during the quotation process to ensure achievable and profitable contracts. The prediction models and generation of accurate digital twins is a collective modeling effort which requires both detailed modelling of the process as well as the dynamic machine behaviour. Considering the machining processes, current literature provides accurate models to predict milling process physics [5, 6, 7]. Nevertheless, when applied in practice, these models show large discrepancies from the actual process behaviour.

One reason can be identified as the influence of the machine tool drive dynamics. In particular, the behaviour of the Numerical Control (NC) plays a key role. Trajectory generation (interpolation) algorithms embedded in the NC system, control the feedrate profile, which is a key input for machining process models. For example, contouring (positioning) errors alter tool engagements [8] which lead to inaccurate force predictions [9]. Thus, in order to accurately develop realistic digital twins for machining processes, the feedrate profile generated by the NC

system of a machine tool must be accurately predicted. This paper deals with modeling and prediction of interpolator dynamics of modern NC systems to accurately estimate machining cycle times and cutting forces along complex parts.

In this paper the term "cycle time" is used to measure the "machining cycle time" which refers to the overall feed motion duration to travel along a machining part program. Once a part program (G-code) is deployed to a CNC machine tool, the NC unit parses the part program and interpolates the tool motion between successive cutter locations (CL). Most modern CAM systems provide tool-paths in terms of discrete CL-data and rely on linear interpolation algorithms that run in the NC units. With the introduction of cheap memory modules, long part programs do not pose a limit, and even basic circular paths are programmed with series of short linear segments [10, 11]. Therefore, modern NC systems are equipped with propriety algorithms that interpolate these lengthy series of short CL-blocks smoothly. These algorithms are called *Look-ahead* or *Compressor* functions and are capable of generating a non-stop motion with time optimal feed-rate profile [12] that respects kinematic limits of the machine [13, 14]. Prediction of a machine's actual feedrate profile requires detailed modeling of the NC system's real-time interpolation behaviour. This includes the motion transition between CL-blocks, for example a typical feedrate profile for continuous motion is shown in Fig. 2.1. During the initial linear motion from zero to commanded feedrate the performance and behavior of the machine tool is dependent upon the acceleration and jerk constraints alone. However, as the tool approaches the end of the first CL-line (corner transition 1 in Fig.2.1) to change the feed direction the tool decelerates to a minimum cornering feedrate before accelerating again to the commanded feedrate. The reduction in feedrate in the vicinity of CL-line junction point is due to both the machine tool satisfying the tool centre point (TCP) error tolerance constraints throughout the cornering transition and the machine tool kinematic constraints [15] . The TCP error can be seen at corner transition 2 where the TCP is maximum displacement between the CL-line and the TCP position. The TCP error constraint imposed upon the toolpath limits the maximum feedrate during cornering transitions and this significantly affects the overall machining cycle time.

Most NC systems utilize jerk limited trajectory generation to smoothly alter feedrate and interpolate along CL-lines [13, 16]. The generated feedrate profile is defined in the form of a cubic polynomial [10]. Axis acceleration limits are imposed based on the torque/power capacity of the drives, and the jerk limits are set to limit unwanted vibrations during rapid feed motion

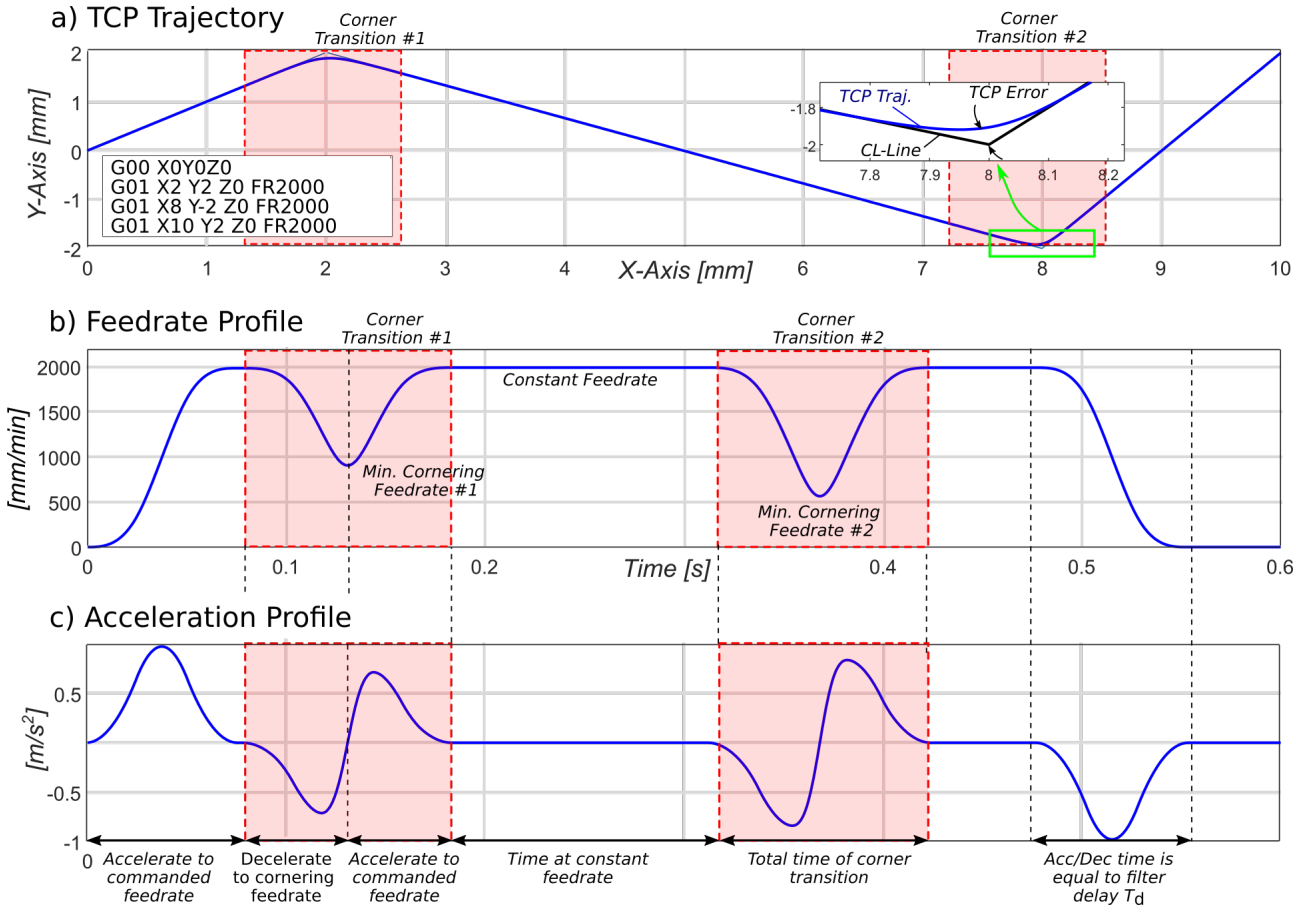

Figure 2.1: Typical Kinematic Profiles of an NC Program.

[17]. This general jerk-limited feedrate profile is well-known, and acceleration and jerk limits of the machine can be read from the NC system. Therefore, the use of jerk limited trajectory as a template allows prediction of feedrate kinematics of modern NC systems and it can correctly predict point-to-point (P2P) trajectories. During P2P interpolation, the tool accelerates from a full-stop to the set feedrate and decelerates again for a full-stop at the end of the CL line. Once the acceleration and jerk limits are known, the feedrate profile can be generated to predict cycle times. Past research considered modeling of NC behaviour of 3 and 5 axis machine tools for both P2P and contouring paths [18, 19, 20].

Predicting feedrate profiles along short segmented complex toolpaths for high speed machining (HSM) is a challenging task. This is due to the fact that look-ahead modules of NC systems alter jerk limits on the fly as it blends series of CL-lines to generate a non-stop smooth continuous feed motion. Here, modeling the path blending behaviour is crucial. NC systems blend linear CL-lines together smoothly while applying geometric blending error and kinematic limit control. Machine tool literature reports that circular arcs [10], cubic [10] or quintic splines [13] can be used for such geometric path blending. There are also methods based on filtering where the discrete toolpath is blended based on low-pass filtering. Finite Impulse Response (FIR) filters are used for such purpose [21]. Such filtering based techniques are more computationally efficient and greatly favored for real-time interpolation on NC systems. For instance Heidenhain [15, 22], Mitsubishi [23] and more recently Siemens [24] NC systems utilize FIR and IIR (infinite impulse response) filters for look-ahead and non-stop smooth interpolation. Typically, users enter a blending tolerance which confines the path blending (contour) errors. Based on the blending tolerance the NC system approximates the given discrete CL-lines and plans the fastest motion with its kinematic limits. Therefore, accurate prediction of cycle times for conventional toolpaths requires modeling of NC system's non-stop interpolation behaviour along linear paths.

This paper models the non-stop interpolation behaviour of modern NC systems and predicts feedrate profiles along HSM toolpaths by considering the real-time path blending behaviour of NC systems. Section 2.2 briefly introduces the low-pass filtering based real-time interpolation method, which is used as a template. It is then used to predict P2P and contouring motion of NC systems in subsequent sections 2.2.3 and 2.3. Illustrative examples and experimental validations are provided in each section. Finally, Section 2.4 provides realistic cycle time, feedrate profile and cutting force prediction for complex aerospace parts.

2.2 Low-pass Filtering Based Real-Time Interpolator Dynamics

This section models real-time interpolation behaviour of an NC system to predict the feedrate profile and overall machining cycle time. Most conventional NC systems utilize IIR or FIR filtering based techniques for computationally efficient real-time interpolation and feed profile planning. In this work, Finite Impulse Response (FIR) filters are used to capture the NC system's behaviour. A simple 1st order FIR filter can be expressed in the Laplace (s) domain by:

$$M_i(s) = \frac{1}{T_i}\frac{1 - e^{-sT_i}}{s}, \quad i = 1 \dots n, \tag{2.1}$$

where s is a complex number, T_i is the time constant of the i^{th} filter. The impulse response is depicted in Fig. 2.2. As seen in (2.1), the filter contains an integrator, which acts to smooth the input signal. These two features of 1st order FIR filters are appealing from a NC system perspective, since G-codes (represented by rectangular velocity pulses) can be convolved through a series of such filters to generate smooth velocity profiles. Since the area underneath the rectangular impulse response is unitary, the area underneath the original input is not altered. [21, 15, 23, 24, 25, 26].

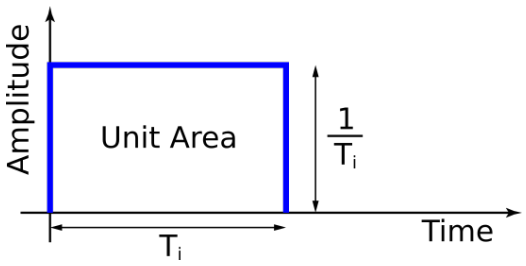

Figure 2.2: Impulse response of a 1st order FIR filter.

Fig.2.4 illustrates this filtering based interpolation procedure. As shown, consider a G-code for a total displacement command of L at a feedrate of F. It is represented by a velocity pulse with an amplitude of F and duration of T_v hence $L = FT_v$. Subsequent convolution of the velocity pulse with the FIR filter yields the higher order velocity response. Using 2-FIR filters in series generates reference trajectories with piece-wise constant jerk profiles and using three FIR filters in series further smooths the reference velocity making them snap limited. Although jerk-limited trajectories are most common in high speed machinery, snap limited trajectories

are tuned for ultra-precision machines [27] to further mitigate the effect of unwanted vibrations. The duration of the original velocity pulse T_v and the time constants of the filters T_n determine the velocity and acceleration profiles, which can be derived analytically by evaluating the convolution integral between the input velocity pulse and the rectangular impulse response of the filter as follows:

$$\begin{aligned} v'(t) =& v(t) * m(t) \\ =& \frac{1}{T_1} \int_0^t \left([v(\tau) - v(\tau - T_v)] \left[u(t-\tau) - u(t - T_1 - \tau) \right] \right) d\tau \\ =& \frac{1}{T_1} \left[\begin{array}{l} \int_0^t v(\tau) u(t-\tau) d\tau - \int_0^t v(\tau) u(t - T_1 - \tau)\, d\tau \\ - \int_0^t v(\tau - T_v)\, u(t-\tau) d\tau + \int_0^t v(\tau - T_v)\, u(t - T_1 - \tau)\, d\tau \end{array} \right] \end{aligned} \tag{2.2}$$

where $v(t)$, $v'(t)$ and $m(t)$ represent the velocity pulse, interpolated velocity signal and the impulse response of the FIR filter (Eq.(2.1) respectively. For multiple first order FIR filtering with different time constants, a detailed analysis is given in [21]. This research introduces linear interpolation using FIR filters with identical time constants.

2.2.1 FIR Interpolation with Matching Time Constants

Typically, the time constants of FIR filters are selected to mitigate structural vibrations of the machine tool [28]. Matching the time constant with the vibration period of the lightly damped modes helps avoid exciting them during rapid acceleration. One method to specify the time constants is to set them equal, $T_1 = T_2$. In this special case, the FIR filter acts as a pure low pass filter with a roll-over frequency of $\omega_c \approx \frac{2\pi}{T_1}$. Fig.2.3 shows the attenuation in the frequency response for multiple FIR filters with matching time constants. The time constant, when set low enough, helps prevent the excitation of any higher frequency vibrations during rapid accelerations. This simpler method compared to tuning individual filters provides a convenient method of vibration suppression during high feedrates.

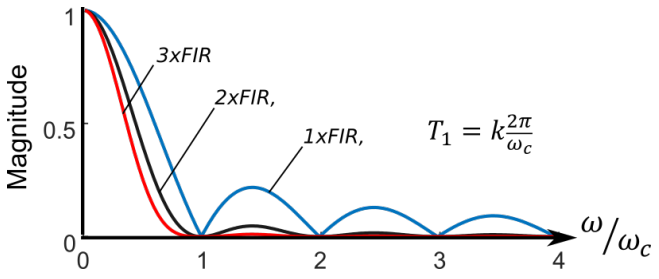

Figure 2.3: Magnitude of the frequency response of multiple FIR filters

For interpolation using 2-FIR filters with matching time constants, the transfer function of the resulting FIR filter is:

$$M_{2FIR}(s) = \left(\frac{1}{T_1}\frac{1-e^{-sT_1}}{s}\right)\left(\frac{1}{T_1}\frac{1-e^{-sT_1}}{s}\right) \tag{2.3}$$

and the resulting velocity profile when a rectangular feed pulse $v(t)$ is filtered $T_1 = T_2 < T_v$ becomes:

$$v'(t) = v(t) * m_{2FIR}(t) \tag{2.4}$$

$$v'(t) = \begin{cases} \frac{F}{2T_1^2}t^2 & 0 \le t < T_1 \\ \frac{F}{2T_1}\left(-t^2 + 4T_1t - 2T_1^2\right) & T_1 \le t < 2T_1 \\ F & 2T_1 \le t < T_v \\ \frac{F}{2T_1}\left(-t^2 + 2T_vt - T_v^2 + 2T_1^2\right) & T_v \le t < T_v + T_1 \\ \frac{F}{2T_1^2}\left(t^2 - 2T_vt - 4T_1t + (T_v + 2T_1)^2\right) & T_v + T_1 \le t < T_v + 2T_1 \end{cases} \tag{2.5}$$

The corresponding acceleration and jerk responses can be derived from (2.5) as:

$$a'(t) = \begin{cases} \frac{F}{T_1^2}t & 0 \le t < T_1 \\ \frac{F}{T_1^2}\left(-t + 2T_1\right) & T_1 \le t < 2T_1 \\ 0 & 2T_1 \le t < T_v \\ \frac{F}{T_1^2}\left(-t + T_v\right) & T_v \le t < T_v + T_1 \\ \frac{F}{T_1^2}\left(t - T_v - 2T_1\right) & T_v + T_1 \le t < T_v + 2T_1 \end{cases} \tag{2.6}$$

$$j'(t) = \begin{cases} \frac{F}{T_1^2} & 0 \le t < T_1 \\ -\frac{F}{T_1^2} & T_1 \le t < 2T_1 \\ 0 & 2T_1 \le t < T_v \\ -\frac{F}{T_1^2} & T_v \le t < T_v + T_1 \\ \frac{F}{T_1^2} & T_v + T_1 \le t < T_v + 2T_1 \end{cases} \tag{2.7}$$

When a square velocity pulse of magnitude F and length T_v is convolved with a first order FIR filter with time constant T_1 the result is a trapezoidal velocity profile with constant acceleration of magnitude F/T_1 (Fig. 2.4b). The total length of the kinematic profiles are extended by the filter time constant T_1 to $T_v + T_1$. When the trapezoidal velocity profile is convolved with a second first order FIR filter with a matching time constant $T_1 = T_2$ the smoothness

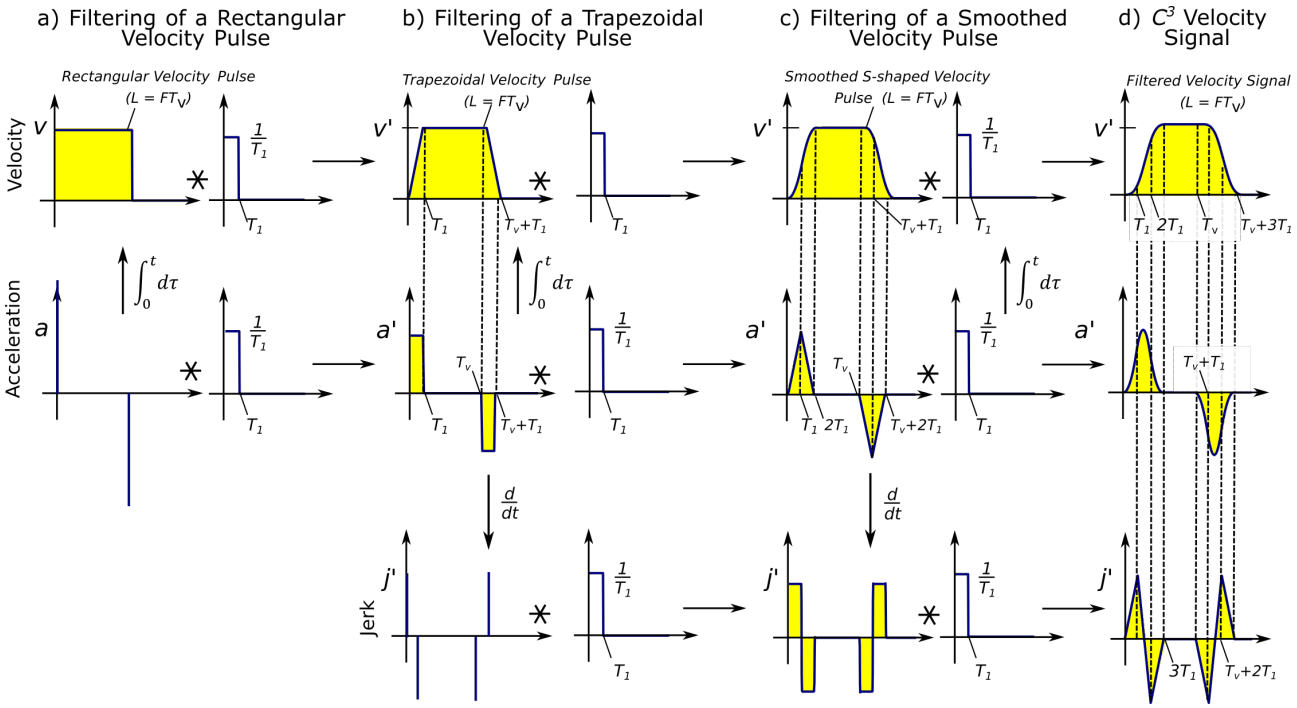

Figure 2.4: Smooth trajectory generation by 3 first order FIR filters with matching time constants

(order) of the velocity profile is increased. The continuity increases from C^1 to C^2, where C^n is the space of n$^{\text{th}}$ order continuously differentiable functions, as shown in equations (2.5) and Fig. 2.4c. However, using the matching time constant $T_1 = T_2$, results in five sections in the kinematic profile and not seven as for the case for two different time constants where $T_1 \neq T_2$. The resulting acceleration profile is triangular around T_1 and $T_v + T_1$ with peak magnitudes F/T_1 and lengths of $2T_1$; the now jerk limited profile has peak magnitudes of F/T_1^2. The total length of the kinematic profiles is extended to $T_v + 2T_1$. The relationship between T_1 and T_v determines the kinematic constraints as for the different filter cases.

Convolving the velocity profile with a third first order FIR filter with the same time constant $T_1 = T_2 = T_3$ results in a C^3 velocity profile, C^2 acceleration profile and C^1 jerk profile. The velocity, acceleration and jerk equations for the 3-FIR case is shown in appendix 2A. The smooth acceleration profile has a peak magnitude of $3F/4T_1$ at times $1.5T_1$ and $T_v + 1.5T_1$ and the jerk profile has peak magnitudes of F/T_1^2. The overall length of the kinematic profiles have been extended from the original square velocity pulse length T_v to $T_v + 3T_1$. The total filter delay when using 3-FIR filters with matching time constants T_1 is therefore $3T_1$.

It can be shown that a high order FIR filter can be accurately modelled and implemented with

using only 3 first order FIR filters. The benefit of using 3 or more first order FIR filters with the same time constant is that the filter response approaches that from a Gaussian filter. The Gaussian response has no overshoot whilst minimising the acceleration and deceleration time periods which makes it the ideal time domain filter for interpolating kinematic profiles [29]. The ability to approximate the Gaussian filter with 3 FIR filters with the same time constant simplifies the design and selection of the filter to a single design parameter T_1. For both the 2 and 3 FIR filter cases, T_1 can be analytically calculated from the maximum permissible jerk J_{max} using Eq.(2.6) and (2.32) respectively as follows:

$$J_{max} = \frac{\Delta F}{T_1^2}, \longrightarrow T_1 = \sqrt{\frac{\Delta F}{J_{max}}} \tag{2.8}$$

2.2.2 Identification of Real-Time Interpolator Dynamics of an NC system

The previous section presented the filtering based real-time trajectory generation. In this section it is shown how the interpolator response of a machine tool can be modelled via the identification of the filter time-constants. A case study was conducted on the DMG Mori eVo40 machine tool shown in Fig.2.5. The machine is commanded by a single G-code to move 6 mm at a speed of 3000 mm/min, and the interpolated reference motion profile is recorded on the NC system directly at a sampling time of $T_s = 0.003s$. Figs 2.5c to 2.5h, show the recorded kinematic profiles. The machine is set to undergo a simple point-to-point (P2P) motion and therefore the tool comes to a full stop before moving to the next commanded position. As shown for the measured system, the NC system generates smooth velocity and acceleration profiles. The acceleration profile mimics a smooth 'bell-shaped' profile. Overall, acceleration, and deceleration duration are measured to be $T_{acc} = T_{dec} = 0.0765$ sec. The cruise velocity portion is roughly measured to be 0.023 sec.

In order to simulate the feed profile, a series of 2 and 3-FIR filters are used. For the 2-FIR case the time constant is selected as $T_1 = \frac{T_{acc}}{2}$ and for the 3-FIR case it is set to $T_1 = \frac{T_{acc}}{3}$. The predicted velocity, acceleration and jerk profiles for the 2-FIR case are shown in Figs 2.5c,2.5e and 2.5h respectively. The time of the measured displacement is equal to the time of the predicted displacement. The difference between the velocity profiles is due to the acceleration. The 2-FIR case exhibits the triangular acceleration profile compared to the smooth measured response. The maximum acceleration for the 2-FIR case is constrained and less than the

measured response.

In order to compare the different filter cases the machine is commanded to move along the same G-code, and the proposed interpolator model for the 3-FIR case is used. As shown in Fig.2.5d the velocity profiles for the 3-FIR case closely resembles the measured velocity profile and the total time of the measured displacement matches the total time for the simulated displacement. The simulated acceleration profile is smooth and the maximum acceleration is higher than for the 2-FIR case but still lower than the measured response. Increasing the order of the simulated system would allow the maximum acceleration to approach the measured response. In general, by increasing the order of the FIR filter, the predicted acceleration profile of the filtered pulse approaches the acceleration profile of the measured response and results in a simulated velocity profile which closely resembles the dynamics of the machine interpolator.

The filter delay is calculated from the jerk (2.8) and the duration of the acceleration phase in each case is equal to the total filter delay. The time constant (filter delay) can be analytically calculated from machine tools' specifications (J_{max}) and therefore kinematic profiles can be generated using FIR filters without the requirement for parameter identification through system testing.

In this section it has been shown that the dynamics of an NC interpolator are increasingly well-approximated by the series combination of identical first-order FIR filters. In addition, the relationship between the parameters of these first-order filters and the resulting interpolator response have been derived.

2.2.3 Multi-Axis P2P Motion Generation

FIR filtering based interpolation of single axis motion was presented in the previous sections. Extending the method to P2P multi-axis linear motion this section describes the process to interpolate kinematic profiles between two points using high order FIR filters.

The start and end positions of a linear G01 command in 3 axes can be represented by $\mathbf{P}_\mathrm{s} = [P_{s,x}, P_{s,y}, P_{s,z}]^T$ and $\mathbf{P}_\mathrm{e} = [P_{e,x}, P_{e,y}, P_{e,z}]^T$, respectively as shown in Fig. 2.6a. The tool displacement L is calculated by taking the Euclidean norm of the vector between the two commanded positions, $L = \|\mathbf{P}_e - \mathbf{P}_s\|_2$.The velocity pulses of each axis (v_x, v_y, v_z) are calculated by multiplying the feed pulse $v(t)$ by the unit velocity vector $\mathbf{u} = (\mathbf{P_e} - \mathbf{P_s})/\|\mathbf{P_e} - \mathbf{P_s}\|_2$.

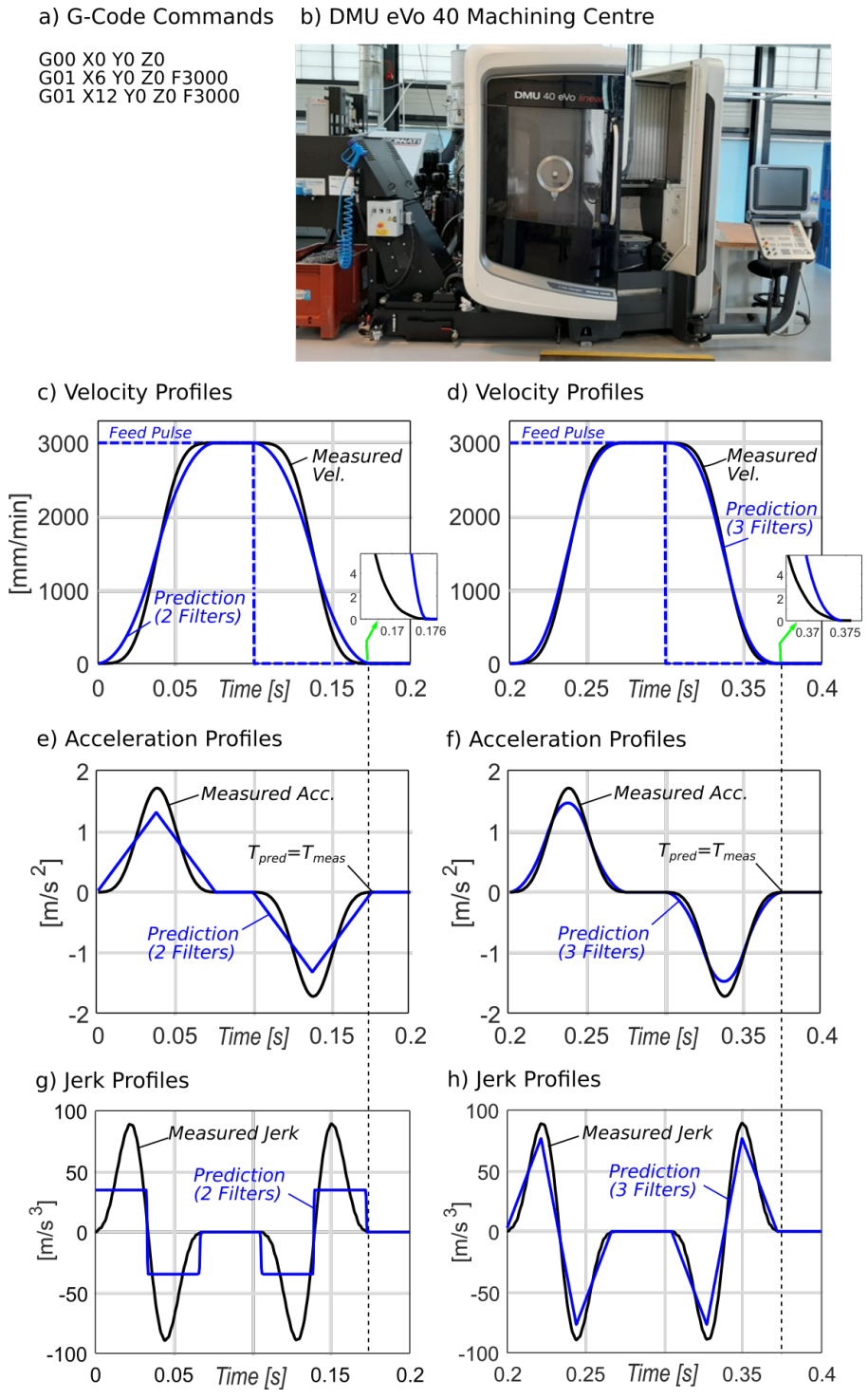

Figure 2.5: Measured and predicted velocity, acceleration and jerk profiles for 2-FIR (c,e,g) and 3-FIR (d,f,h) P2P motion interpolation

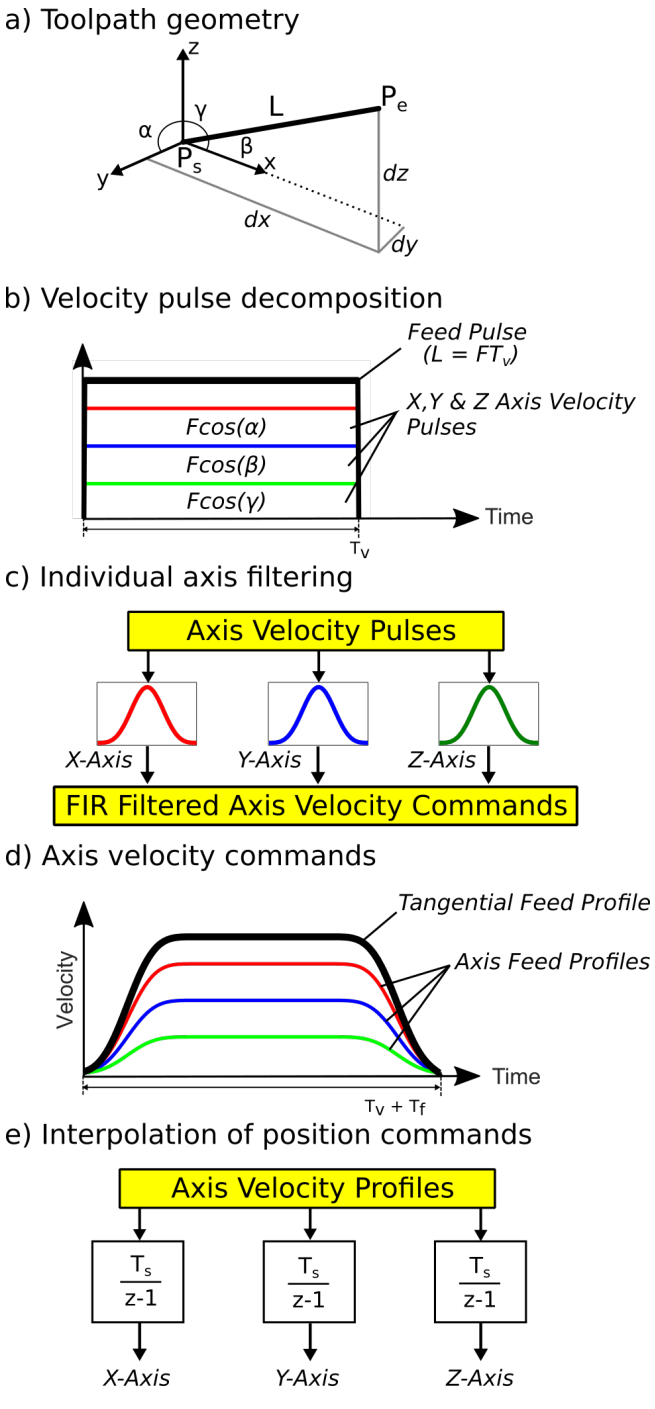

Figure 2.6: Multi-axis interpolation based on high order FIR filtering.

$$\frac{d\mathbf{P(t)}}{dt} = \dot{\mathbf{P}}(\mathbf{t}) = v(t)\mathbf{u} = \begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} \tag{2.9}$$

where $\dot{\mathbf{P}}(\mathbf{t})$ represents the first time derivative of the P2P displacement (Fig. 2.6b).
In order to generate (and interpolate) the reference velocity commands (v'_x,v'_y,v'_z), the individual axis velocity pulses (v_x, v_y, v_z) are convolved with the FIR filter (Figs. 2.6c and 2.6d):

$$\frac{d\mathbf{P'}(t)}{dt} = \dot{\mathbf{P}}'(t) = \begin{bmatrix} v'_x(t) \\ v'_y(t) \\ v'_z(t) \end{bmatrix} = \dot{\mathbf{P}}(t) * m(t) \tag{2.10}$$

Finally, the filtered position commands are generated by integrating the filtered axis velocity commands:

$$\mathbf{P}'(t) = \begin{bmatrix} p'_x(t) \\ p'_y(t) \\ p'_z(t) \end{bmatrix} = \int_0^t \begin{bmatrix} v'_x(t) \\ v'_y(t) \\ v'_z(t) \end{bmatrix} d\tau \tag{2.11}$$

2.3 Prediction of Interpolator Behaviour during Non-stop Motion

The previous section showed that P2P linear interpolation behaviour of an NC system can be modelled by velocity pulses low pass filtered by a series of first order FIR filters. The only required parameter to predict the machine's feed profile and accurately estimate the resulting cycle time is the time constant, i.e. total delay of the FIR filter. As shown, the filter time delay can be calculated from the maximum permissible jerk (2.8) and commanded feedrate. This section focuses on accurate prediction of interpolator behaviour during non-stop contouring motion, which is the most commonly used interpolation technique for high speed machining (HSM).

2.3.1 Modeling of Non-stop (Contouring) Interpolation Behaviour

Typical high speed machining toolpaths found in die and mould manufacturing or in aerospace industry consist of series of short segmented toolpaths [30]. When interpolated in HSM mode, the NC interpolator does not undergo a full-stop at the end of each CL line. Instead, the CL

lines are blended together for a non-stop smooth motion interpolation where machining feedrate is reduced to a cornering speed V_c around junction points of the CL-blocks (See Fig.2.1). The prediction of V_c is crucial to accurately capture the actual feedrate profile and estimate the resultant cycle time. Several constraints affect the cornering speed (V_c) and overall acceleration profile around the CL data points. Firstly, V_c is controlled by the blending (cornering) tolerance [19]. Typically, lower blending tolerance delivers more accurate motion but generates slower feed profiles. In contrary, a larger tolerance value allows faster speeds and shorter overall cycle time. The relationship between the blending tolerance and the feed drop around the corner must be captured. Secondly, the deceleration/acceleration profile and the transition duration from the programmed feedrate (F) to the cornering speed (V_c) are dictated by acceleration and jerk limits of the machine. Both of these key characteristics must be modelled to accurately predict the varying feedrate profile along HSM tool-paths.

In an effort to accurately model the interpolator behaviour, the feed pulse distribution shown in Fig. 2.7b is proposed in this manuscript. Notice that the feed pulse profile is different from the case used for the P2P motion. Feed pulses of each CL block are commanded back-to-back with no dwell time in between. In other words, they are constructed as a continuous pulse stream. The duration of the feed pulse is T_v. Notice that the feed pulse does not have a constant amplitude of F. Instead, around CL block junctions the feed command value is dropped down to F_c. Such small feed pulse is added to model the blending kinematics, commanding the feedrate to drop down to a cornering feed of F_c. The duration of the cornering feed pulse is set to T_b, which controls how long the deceleration and acceleration last around the blend.

When the feed pulse profile is interpolated with a FIR filter the resulting velocity profiles are smooth velocity profiles that better approximate the actual velocity profiles of the machining interpolator. Fig.2.7a and Fig.2.7c show the toolpath and the corresponding interpolated X-axis and Y-axis velocity profiles respectively. The total length of the velocity profiles is equal to the sum of the pulse lengths plus the filter delay T_d. Fig.2.7d shows the cornering feedrate V_c of the resultant velocity profile is equal to the commanded blending pulse feedrate F_c and this occurs at half the filter delay $T_d/2$ from the start of the Y-axis profile.

The cornering feedrate is controlled by setting the blending velocity pulse F_c equal to the desired cornering tangential velocity V_c and setting the acceleration and deceleration time for the interpolated feed profile equal to the time required to reduce from F to F_c. A scaling factor

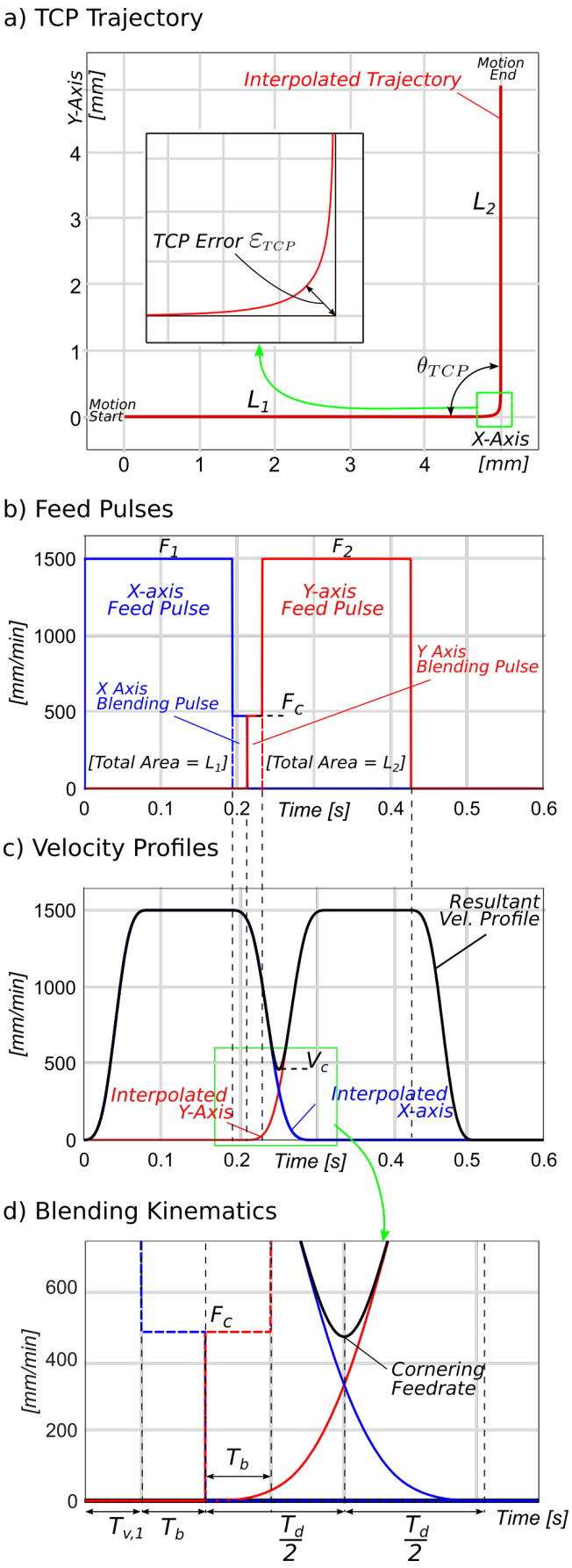

Figure 2.7: FIR based interpolation of a right angled toolpath with a constant feedrate

is applied to F to represent F_c as a function of commanded feedrate F:

$$F_c = F\alpha = V_c \tag{2.12}$$

where V_c is the resultant 3-axis TCP velocity defined as

$$V_c = \sqrt{v_x'^2 + v_y'^2 + v_z'^2} \tag{2.13}$$

and v_x', v_y' and v_z' represent the interpolated axis velocities at the minimum corning feedrate. The total acceleration and deceleration time of the interpolated feed profile to reach $F\alpha$ from F is represented by T_b, it is a function of the filter delay T_d, and it can be calculated as:

$$T_b = \frac{1}{2} T_d \, (1 - \alpha) \tag{2.14}$$

The final task when modeling the pulse train is calculating the main velocity pulse lengths T_v. In section 2.2 the length of the velocity pulse T_v was calculated from L/F, however, with the introduction of the blending pulses, T_v must be modified in order to preserve the total area of the pulses and hence the TCP displacement.

The commanded TCP displacement is calculated from the total area of the velocity pulse and the blending pulse, this can be seen in Fig.2.7b where the total area within the X-axis and Y-axis pulses is equal to L_1 and L_2 respectively. For a single axis displacement L the pulse areas comprise of the main pulse (calculated as FT_v) and the blending pulse (calculated as F_cT_b):

$$L = FT_v + F_cT_b \tag{2.15}$$

Rearranging equation (2.15) and incorporating equation (2.12) yields the modified value of T_v as:

$$T_v = \frac{L}{F} - \alpha T_b \tag{2.16}$$

Equation 2.16 holds for velocity commands with a single blending pulse, this is the case for the initial and final CL lines in a part program which start and end at zero feedrate (full stop). The remaining displacements in a part program are continuous and therefore the commands consist of a velocity pulse with a blending pulse either side as shown in Fig.2.8. Accordingly, each cornering blend consists of two back to back blending pulses.

For the entire pulse train, each G01 command or CL-line can be represented by an index k with k=1 corresponding to the initial command in the part program. The associated feedrate commands in the part program are hence denoted $F(k)$. Therefore, for the main commands in a part program the modified value of T_v is calculated as:

$$T_v(k) = \frac{L(k)}{F(k)} - \alpha(k)T_b(k) - \alpha(k+1)T_b(k+1) \tag{2.17}$$

For constant feedrate the adjoining blending pulses are symmetric. This leads to symmetrical interpolated velocity profiles and results in symmetrical displacement profiles, translating to the same toolpath trajectory for both forward and backward passes resulting in a more accurate finish.

2.3.2 Filtered Signal Generation

The composition of the velocity pulses and filtered kinematic profiles was shown in the previous section. In practise, the strategy for interpolation of multi-segmented NC tool-paths using high order FIR filtering, as shown in Fig. 2.8, is as follows:

1. Read NC code and parse commanded X,Y, Z positions and feedrate commands from individual G01 commands and extract defined tolerance setting.

2. Calculate toolpath geometry (cornering angles θ_{TCP}) and unit velocity vectors for each G01 command.

3. Calculate cornering feedrates where $V_c = F_c = F\alpha$ from the maximum permissible feedrate for the cornering angle and defined tolerance (2.29)(2.30). (demonstrated in section 2.3.3)

4. Calculate pulse velocities F and velocity blending pulse widths T_b followed by modified velocity pulse widths T_v (2.16)(2.17).

5. Synchronise timed axis velocity pulses and generate unfiltered axis velocity signals (pulse train).

6. Define FIR filter time constants for the commanded feedrate from maximum permissible jerk (2.8).

7. Using high order FIR filtering with matching time constants interpolate the axis velocity pulse signals to generate smooth kinematic profiles for each axis (2.10).

8. Finally, integrate the filtered velocity signals to generate synchronised accurate position commands in the time domain (2.11).

Sections 2.3.1 and 2.3.2 described the components of the velocity pulse train and application of FIR filtering for generation of kinematic profiles for non-stop high speed motion. The following sections will analytically demonstrate the relationship between the cornering speed V_c to the blending error and axis kinematic limits and ultimately demonstrate how F_c is selected to guarantee these constraints are satisfied.

2.3.3 Kinematic Profiles for the 2-FIR Filter Case

The geometry of velocity blending pulses was presented and calculated in section 2.3.1. The pulse signals are interpolated using FIR filters to generate kinematic profiles that control the cornering feedrate. This section analytically derives the equations for the kinematic profiles when using velocity blending pulses and FIR filtering based interpolation to control the cornering feedrate. In doing so, the authors are able to analytically calculate the blending pulse feedrate command F_c which satisfies both TCP error and machine kinematic constraints during cornering transitions.

Using 2-FIR filters with matching time constants to interpolate a velocity pulse signal results in the kinematic profiles shown in figure 2.9. The profiles are split into 5 sections during acceleration/deceleration as shown in Fig.2.9b for the Y-axis acceleration. The objective of the analytical expressions is to calculate the interpolated displacement at the point of maximum TCP error and the interpolated velocity at the minimum cornering feedrate. This occurs at half the total filter delay $T_d/2$ (see Fig.2.9a). The total filter delay for the 2-FIR case is $T_d = 2T_1$, where T_1 is calculated from the maximum permissible jerk (equation (2.8)), resulting in the maximum TCP error and minimum cornering feedrate occuring at T_1. Fig.2.9b shows T_1 is at the start of section 3, therefore only sections 1-3 of the kinematic profiles need considering. The analytical expressions for sections 1-3 of the displacement, velocity, acceleration and jerk profiles for the 2-FIR case are presented in equations (2.18) to (2.21) respectively.

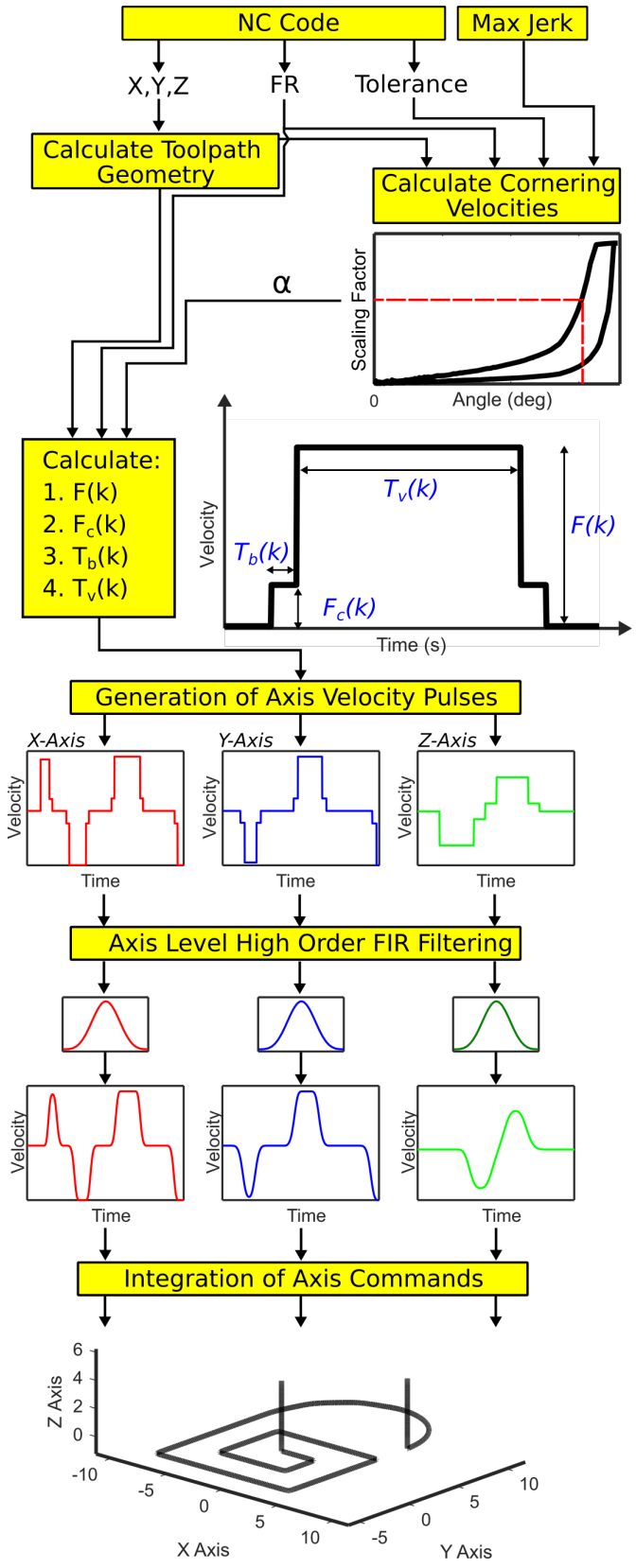

Figure 2.8: Non-stop interpolation of kinematic profiles using high order FIR filtering

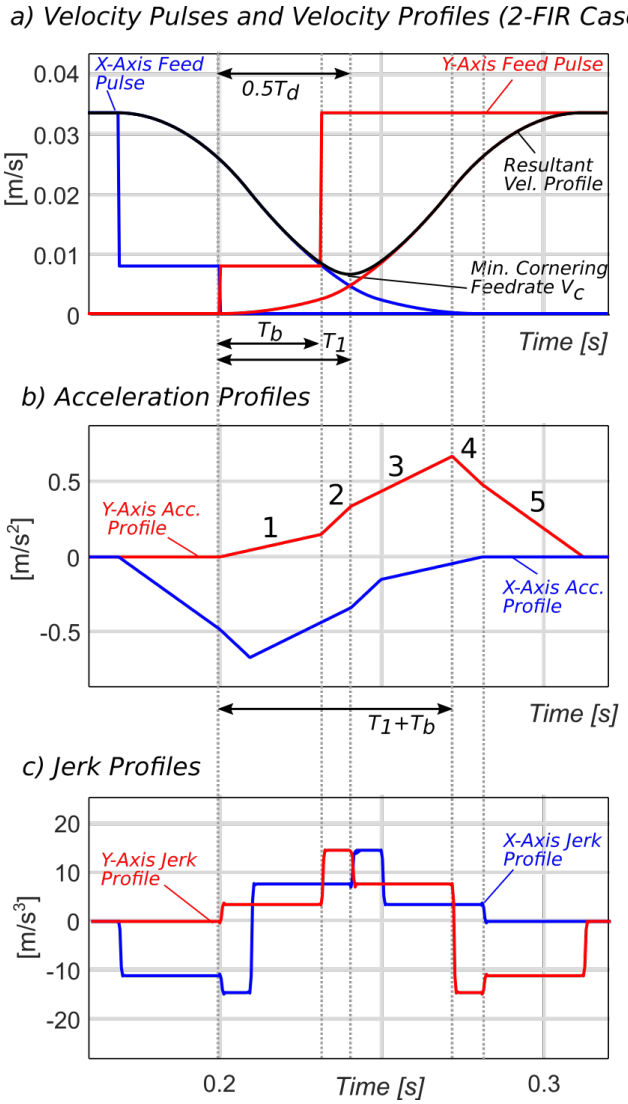

Figure 2.9: Velocity, acceleration and jerk profiles generated by blended velocity pulses interpolated 3-FIR filters

$$s'(t) = \begin{cases} \frac{1}{6}\frac{\alpha F}{T_1^2}t^3 & 0 \le t < T_b \\ \frac{1}{6}\frac{F}{T_1^2}\left(t^3 + 3T_b\left(\alpha-1\right)t^2 + 3T_b\left(1-\alpha\right)t + T_b^3\left(\alpha-1\right)\right) & T_b \le t < T_1 \\ \frac{1}{6}\frac{F}{T_1^2}((1-2\alpha)\,t^3 + (3T_b\left(\alpha-1\right) + 6T_1\alpha)\,t^2 + (3T_b^2\left(1-\alpha\right) + \ldots & \\ \ldots - 6T_1^2\alpha)t + T_b^3\left(\alpha-1\right) + 2T_1^3\alpha) & T_1 \le t < T_b + T_1 \end{cases} \tag{2.18}$$

$$v'(t) = \begin{cases} \frac{1}{2}\frac{\alpha F}{T_1^2}t^2 & 0 \le t < T_b \\ \frac{1}{2}\frac{F}{T_1^2}\left(t^2 + 2T_b\left(\alpha-1\right)t - T_b^2\left(\alpha-1\right)\right) & T_b \le t < T_1 \\ \frac{F}{2T_1^2}((1-2\alpha)\,t^2 + (T_1\alpha + 2T_b\left(\alpha-1\right) + 4T_1\alpha)\,t + T_1^2\alpha + \ldots & \\ \ldots + T_b^2\left(1-\alpha\right) - 2T_1\alpha) & T_1 \le t < T_b + T_1 \end{cases} \tag{2.19}$$

$$a'(t) = \begin{cases} \frac{\alpha F}{T_1^2}t & 0 \le t < T_b \\ \frac{F}{T_1^2}\left(t + T_b\left(\alpha-1\right)\right) & T_b \le t < T_1 \\ \frac{F}{T_1^2}\left(\left(1-2\alpha\right)t + 2T_1\alpha - T_b + T_b\alpha\right) + & T_1 \le t < T_b + T_1 \end{cases} \tag{2.20}$$

$$j'(t) = \begin{cases} \frac{\alpha F}{T_1^2} & 0 \le t < T_b \\ \frac{F}{T_1^2} & T_b \le t < T_1 \\ \frac{F}{T_1^2}\left(1-2\alpha\right) & T_1 \le t < T_b + T_1 \end{cases} \tag{2.21}$$

The interpolated axis velocity at maximum TCP error (minimum cornering feedrate) occurs at $t = T_d/2 = T_1$, therefore in the 2-FIR filter case this results in the following expressions for interpolated velocity (2.22) and displacement (2.23):

$$v' = \frac{1}{2}\frac{F}{T_1^2}\left(T_1^2 - T_b^2\left(1-\alpha\right) + 2T_1T_b\left(\alpha-1\right)\right) \tag{2.22}$$

$$s' = \frac{1}{6}\frac{F}{T_1^2}\left(T_1^3 + 3T_1T_b^2\left(1-\alpha\right) + 3T_1^2T_b\left(\alpha-1\right) + T_b^3\left(\alpha-1\right)\right) \tag{2.23}$$

Using equation (2.14), the interpolated displacement (2.22) and velocity(2.23) can be expressed in terms of F and α:

$$v' = \frac{F}{2}\,\alpha\,\left(-\alpha^2 + \alpha + 1\right) \tag{2.24}$$

$$s' = \frac{F}{6}\,T_1\,\alpha\,\left(-\alpha^3 + \alpha^2 + 1\right) \tag{2.25}$$

Fig.2.10 shows a cornering transition between two CL-lines or G01 commands. The maximum TCP contouring or corner blending error ε_{TCP} occurs in the centre of the cornering trajectory

and is calculated by evaluating the interpolated axis displacements s' at $t = T_1$. The interpolated axis displacements are calculated from (2.25) and the vectors from the corner transition to these positions are represented by $\mathbf{l_1}$ and $\mathbf{l_2}$.

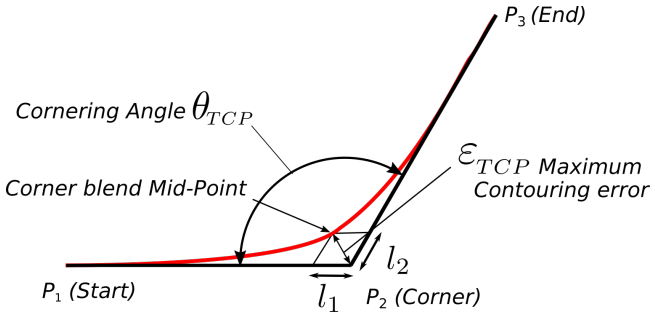

Figure 2.10: Toolpath showing contouring error and cornering angle between two consecutive G01 commands

The contouring error ε_{TCP} (shown in Fig.2.10) is calculated from the Euclidean distance between the vectors $\mathbf{l_1}$ and $\mathbf{l_2}$.

$$\varepsilon_{TCP} = \|\mathbf{l_2} - \mathbf{l_1}\| = \sqrt{l_1^2 + l_2^2 + 2l_1 l_2 \cos\theta_{TCP}} \tag{2.26}$$

where θ_{TCP} represents the TCP cornering angle. Assuming constant feedrate in this example, $l_1 = l_2 = l_\varepsilon$, in which case, (2.26) simplifies to the following expression:

$$\varepsilon_{TCP}^2 \leq 2l_\varepsilon^2 \left(1 + \cos\theta_{TCP}\right) \tag{2.27}$$

Inserting (2.14) and (2.23) into (2.26) enables the TCP corner blending error to be defined as:

$$\varepsilon_{TCP} = \frac{\sqrt{2}}{6} \sqrt{F^2 {T_1}^2 \alpha^2 \left(\cos\theta_{TCP} + 1\right) \left(-\alpha^3 + \alpha^2 + 1\right)^2} \tag{2.28}$$

Using equation 2.28 the TCP error can be calculated for any toolpath geometry and commanded feedrate. The kinematic profiles for the 3-FIR case are shown in Fig.2.17 in appendix 2B and the derivation of TCP error for the 3-FIR case is included in appendix 2C.

To ensure minimum cycle times the actual feedrate must remain as close to the commanded feedrate as possible throughout the toolpath including cornering transitions. However, to satisfy both jerk and TCP error constraints there is a maximum permissible cornering feedrate. Using equations (2.28) and (2.42), it is possible to calculate the relationship between TCP error, maximum permissible cornering feedrate and cornering angle for the 2-FIR and 3-FIR filter cases respectively.

Rearranging equation (2.28), the maximum permissible cornering feedrate for the 2-FIR filter case must satisfy:

$$FT_1\sqrt{\cos\left(\theta_{TCP}+1\right)}\left(\alpha^4-\alpha^3-\alpha\right)-3\sqrt{2}\varepsilon_{TCP}\leq 0 \tag{2.29}$$

and for the 3-FIR filter using (2.42) the maximum permissible cornering feedrate must satisfy:

$$FT_1\sqrt{\cos\left(\theta_{TCP}+1\right)}\left(16\alpha^5+16\alpha^4-8\alpha^3-16\alpha^2-85\alpha-1\right)-192\sqrt{2}\varepsilon_{TCP}\leq 0 \tag{2.30}$$

For a commanded feedrate F and range of cornering angles $\theta_{TCP}\in[0°,180°]$, equations (2.29) and (2.30) are solved for solutions $0\leq\alpha\leq 1$ to calculate the limit to the feedrate scaling factor α. When multiplied by the commanded feedrate F this represents the maximum permissible cornering feedrate that can be achieved whilst satisfying the kinematic and tolerance constraints. The blending pulse feedrate F_c is commanded to this limit value.

The reduction in cornering feedrate for both the 2-FIR and 3-FIR filter cass are shown in Fig.2.11. Cornering feedrates selected below the curves will satisfy the TCP error constraints for the commanded feedrate and cornering angle. The figure shows the limits for 10μm and 50μm tolerance constraints. For the 50μm tolerance, the figure shows higher cornering feedrates can be achieved compared to the 10 μm case. It can also be recognised that higher feedrates can be achieved in the 3-FIR. Therefore there is an advantage of using 3-FIR filters to reduce the overall machining cycle time as the tool can remain at higher feedrates during cornering transitions than for the 2-FIR case. Despite the advantage of using a higher order filter, there remains a limit to the order of filters that can be used effectively for trajectory generation. As the order is increased the filter time constant reduces. In the frequency domain the notch (as shown in Fig.2.3) will shift to higher frequencies. This will be constrained by the lowest structural mode of the machine tool.

This section has shown a method of using multiple first order FIR filters with matching time constants to model continuous linear interpolation of velocity pulse signals. It has been show that the cornering feedrate and TCP error can be controlled using velocity blending pulses. This method has been extended to predict feedrates and machining cycle time for toolpaths of any geometry and defined TCP tolerance. The following section demonstrates and validates the proposed method on industrial case studies.

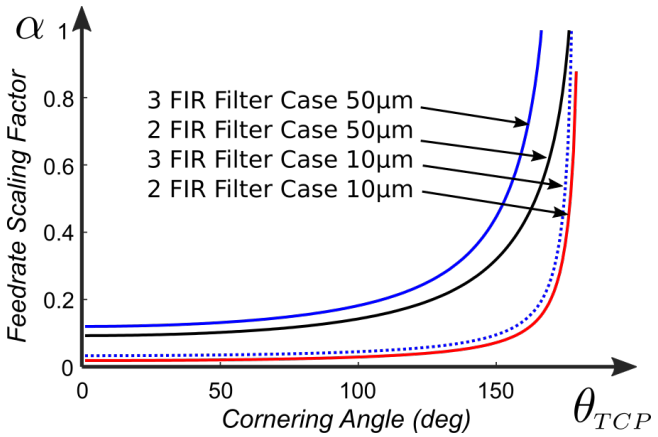

Figure 2.11: Minimum cornering feedrate and cornering angle curves shown for 10μm and 50μm tolerance settings at 2000 mm/min for both 2 and 3 first order FIR cases

2.4 Experimental Validation

Machining experiments were conducted on a DMG Mori Universal eVo 40 5-axis machining centre with a Heidenhain TNC640 controller. Two short tool-paths were used for pocketing operations and a single long aerospace part program is evaluated in the cycle time prediction. The section concludes with validation of the feedrate prediction method integrated with a virtual machining model.

2.4.1 Case Studies on Pocketing Toolpaths

The first two case studies, as shown in Fig.2.12, consist of a contour and a trochoidal pocketing tool-path. These tool-paths are generated by CAM software [31] and the part programs are deployed to the machine directly with no modification. Table 2.1 shows the cutting conditions. As noted, 2 different feedrates 1000 and 3000 mm/min are used. The most important setting is contour error tolerance for HSM. Two different contouring tolerance, 10 and 50 μm are used. Table 2.1 summarises the cycle time results. All simulated trajectories in the case studies were modelled using the method described in section 2.3.2 and 3-FIR filters.

Machining Cycle Time Estimation

The predicted machining cycle times are compared with the measured CNC and CAD/CAM calculated machining cycle times. The results are presented in Table 2.1 and Fig.2.13a. For all cases the predicted machining cycle times are accurate to within 3% of the measured cycle time with the exception of the trochoidal pocket (1000 mm/min, 10μm case) which is 5.52%. These compare favourably to the CAD/CAM calculated cycle times which has an error range from

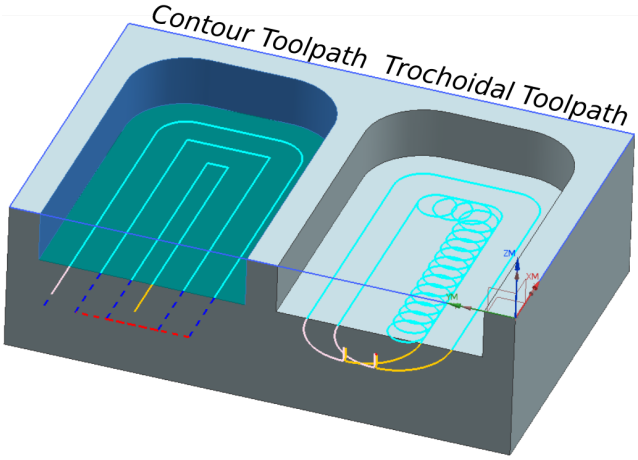

Figure 2.12: Contour (left) and Trochoidal (right) pocketing toolpaths designed in Siemens NX CAM

0.22% to 54.99%. The significant result is the Trochoidal pocket (3000 mm/min and 10μm case). The proposed method is able to accurately predict the increase in machining cycle time from 14.40 to 30.99 seconds when tightening the tolerance from 50μm to 10μm, which is within 2.72% of the measured cycle time. This is compared to an error of 54.99% for the CAD/CAM calculated cycle time.

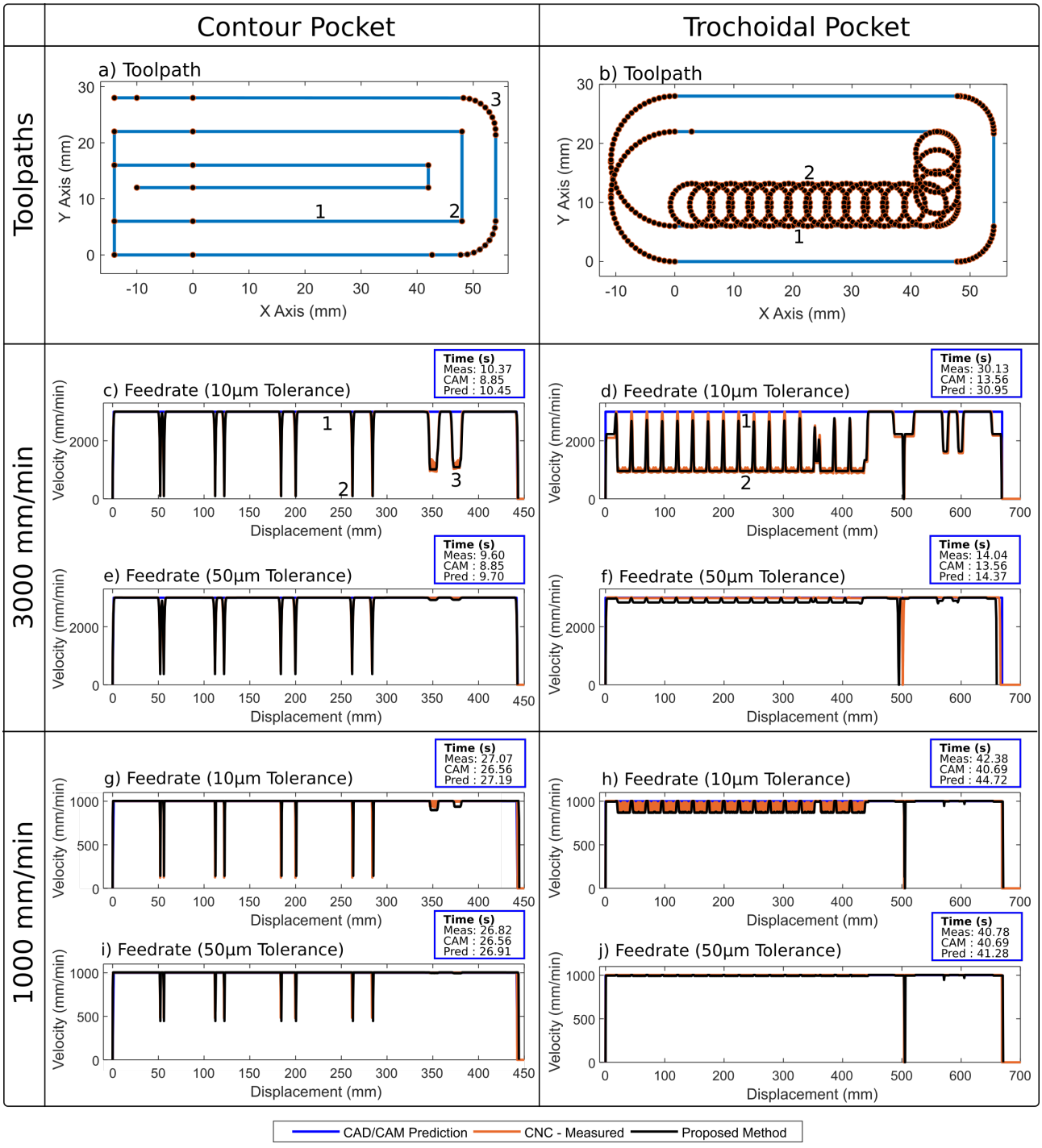

Figure 2.13: Pocketing case studies - predicted, measured and CAD/CAM tangential velocities

Case Study	**Feedrate (mm/min)**	**Tolerance (microns)**	**Measured Time (sec)**	**CAD/CAM Time (sec)**	**FIR Predicted Time (sec)**	**CAD/CAM Prediction Error (%)**	**FIR Prediction Error (%)**
Contour Pocket	1000	10	27.07	26.56	27.32	-1.88	0.92
Contour Pocket	1000	50	26.82	26.56	26.95	-0.97	0.48
Contour Pocket	3000	10	10.37	8.85	10.46	-14.66	0.87
Contour Pocket	3000	50	9.6	8.85	9.73	-7.81	1.35
Trochoidal Pocket	1000	10	42.38	40.69	44.75	-3.98	5.59
Trochoidal Pocket	1000	50	40.78	40.69	41.30	-0.22	1.28
Trochoidal Pocket	3000	10	30.13	13.56	30.99	-54.99	2.86
Trochoidal Pocket	3000	50	14.04	13.56	14.40	-3.41	2.56

Table 2.1: Machining cycle times for contour and trochoidal pocket at different machining parameters

Prediction of Feedrate Profile

To demonstrate the performance of the feedrate prediction method a number of toolpath features were selected. The predicted, CAD/CAM calculated and measured CNC tangential velocities at these particular features were recorded and are presented in Tables 2 and 3. The contour pocket features consist of (1) a long G01 segment, (2) a sharp corner and (3) a rounded corner consisting of small G01 segments. The trochoidal pocket features consist of (1) the stepover segment and (2) the main arc. Depending on the tolerance and the commanded feedrate large differences in tangential velocity can exist between the stepover segment and the main arc of a trochoidal toolpath which in turn results in a large cyclical variation of cutting forces. It is for this reason they are included in this study. The features described above are shown on the toolpaths in Fig. 2.13 and the corresponding position with respect to displacement and tangential velocities are demonstrated directly beneath. Overall, the prediction error ranges from 0.1-10.3% compared with CAD/CAM calculated error range of 0.22-2555%, where the error is calculated as a percentage difference from, and with respect to, the measured tangential velocity. The performance of the proposed feedrate prediction method at each feature is described below:

Long G01 Segment. The prediction error range is between 0.1-0.13% compared to the CAD/CAM calculated error range of 0.27-0.3%. The high accuracy is to be expected as no feedrate limiting features are present in the segment. The difference in measured velocity compared to the idealised CAD/CAM values are due to interpolator rounding during trajectory generation.

Sharp Corner. The prediction error range is between 7-20% compared to the CAD/CAM calculated error range of 107-2555%. The fundamental difference is due to the CAD/CAM calculation not taking into account the cornering kinematic constraints due to tolerance and thus not predicting the reduction in feedrate during the cornering segment. This holds true for all of the features demonstrated except the long G01 segment. For the 10μm tolerance cases the tool comes to an almost complete stop - 4% and 11% of the commanded feedrate for the 3000 mm/min and 1000 mm/min cases respectively, the presented method predicts these reductions.

Rounded Corner. The prediction error range is between 0.3-5.4% compared to the CAD/CAM calculated error range of 0.2-168%. The significant result is the 3000 mm/min and 10μm case (Fig. 2.13c) where the CAD/CAM calculation does not account for the reduction in velocity

due to the tolerance requirement. The CAD/CAM calculated error is 168% compared to the measured value and the prediction error is within 2.4%.

Trochoid Stepover. The prediction error range is between 0.1-10% compared to the CAD/CAM prediction error range of 0.2-0.5%. The CAD/CAM calculation does not predict any differences along the trochoidal toolpath between the stepover and the main arc. This can be seen in Fig. 2.13d for the 3000 mm/min 10μm case. The blue line shows the CAD/CAM prediction but the actual kinematic profile is very different. The stepover results in tangential velocities close to the commanded feedrate as the cornering angles between the segments are less acute than for the rest of the main arc.

Trochoid Main Arc. The prediction error range is between 0.6-7% compared to the CAD/CAM calculated error range of 0.2-208%. The reduction in tangential velocity around the main arc is due to the cornering angles between the segments. The influence of the toolpath tolerance on the cornering tangential velocity can be seen in Fig.2.13d and Fig.2.13f. The increase in tolerance from 50μm to 10μm results in more than a 65% reduction in tangential velocity around the main arcs of the trochoids. The prediction method accurately predicts the feedrate within 1.5% of tangential velocity measured at the main arc. Taking this result one step further, this demonstrates that a feedrate driven cutting force model when incorporating the prediction method will be able to predict the cyclical cutting forces due to the 65% variation in magnitude of feedrate fluctuations around the trochoidal toolpath.

Case Study	**Feedrate (mm/min)**	**Tolerance (microns)**	**Analysis Point**	**Tangential Velocities (mm/min)**				
				Measured	**CAD/CAM**	**Proposed Prediction**	**CAD/CAM Error**	**Proposed Prediction Error**
Contour	1000	10	Point 1 Straight G01	1003	1000	1004	-0.30%	+0.10%
Contour	1000	50		1003	1000	1004	-0.30%	+0.10%
Contour	3000	10		3008	3000	3005	-0.27%	-0.10%
Contour	3000	50		3008	3000	3004	-0.27%	-0.13%
Contour	1000	10	Point 2 Sharp Corner	124	1000	143	+706%	+15%
Contour	1000	50		482	1000	446	+107%	-7%
Contour	3000	10		113	3000	93	+2555%	-17%
Contour	3000	50		465	3000	370	+545%	-20%
Contour	1000	10	Point 3 Rounded Corner	994	1000	940	+0.6%	-5.4%
Contour	1000	50		998	1000	995	+0.2%	-0.3%
Contour	3000	10		1120	3000	1093	+168%	-2.4%
Contour	3000	50		2996	3000	2929	+0.13%	-2.23%

Table 2.2: Contour pocket case study: tangential velocity prediction and performance

Case Study Pocket	**Feedrate (mm/min)**	**Tolerance (microns)**	**Analysis Point**	**Tangential Velocities (mm/min)**				
				Measured	**CAD/CAM**	**Proposed Prediction**	**CAD/CAM Error**	**Proposed Prediction Error**
Trochoidal	1000	10	Point 1 Trochoid Stepover	1005	1000	1003	-0.50%	0.20%
Trochoidal	1000	50		1005	1000	1004	-0.50%	0.10%
Trochoidal	3000	10		2994	3000	2687	+0.20%	10.3%
Trochoidal	3000	50		3014	3000	2985	-0.46%	0.96%
Trochoidal	1000	10	Point 2 Trochoid Main Arc	938	1000	872	+6.61%	7%
Trochoidal	1000	50		1002	1000	994	-0.20%	0.60%
Trochoidal	3000	10		973	3000	958	+208%	1.5%
Trochoidal	3000	50		2986	3000	2838	+0.47%	4.96%

Table 2.3: Trochoidal pocket case study: tangential velocity prediction and performance

2.4.2 Case Study 3 - Aerostructure Toolpath

An industrial toolpath was chosen to validate the method against a representative aerostructure part. The part program consists of three toolpaths - roughing, finishing #1 floors and finishing #2 walls as shown in Fig. 2.14. The part programs were run at three tolerance settings, 10μm, 20μm and 50μm to demonstrate the significant impact tolerance has on machining cycle times and therefore on feedrate and cycle time prediction. Table 2.4 compares the predicted machining cycle times with both the measured cycle times and the predicted times from a commercial CAD/CAM software package for each individual toolpath.

The overall machining cycle times, calculated by summing the cycle times for the 3 sections of the part program, are shown in Table 2.5. The CAD/CAM prediction error ranges from 62.41% under prediction for the 10μm case to 36.42% under prediction for the 50μm case. The actual CAD/CAM predicted times do not change as the software does not account for tolerance, the calculation is based upon distance travelled along the toolpath and ideal feedrate. Therefore as the tolerance is relaxed the measured cycle time approaches the CAD/CAM case and their prediction becomes more accurate.

The prediction error from the proposed method (as shown in Table 2.5) ranges from 3.50% over prediction for the 10μm case to 4.69% for the 50μm case. The 20μm case has a prediction error of 5.34% under the measured cycle time which is approximately 10% of the CAD/CAM error (51.49%) for that particular case. The aerostructure case study validates the model for predicting both feedrate and machining cycle times for varying tolerance settings on very complex industrial toolpaths far outperforming the CAM software.

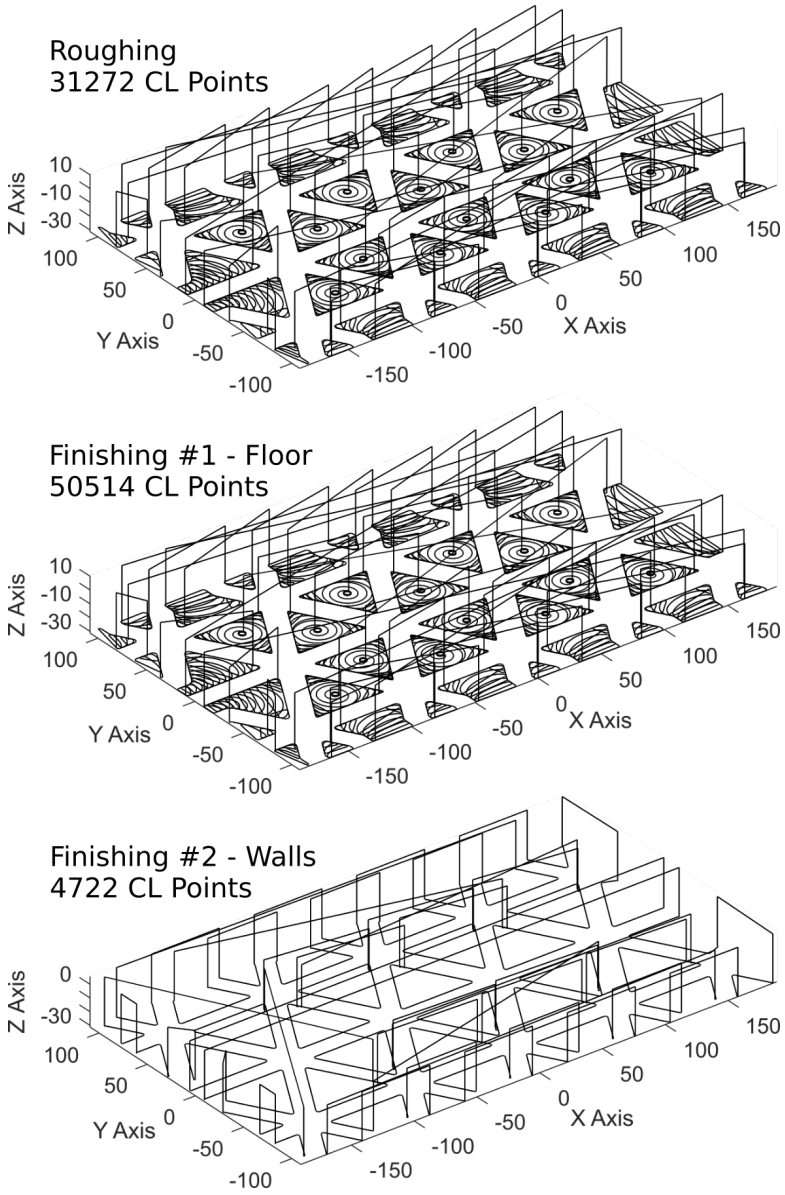

Figure 2.14: Aerostructure toolpaths (shown in order of operation)

Section	**Feedrate (mm/min)**	**Tolerance (microns)**	**Measured Time (sec)**	**CAD/CAM Time (sec)**	**Proposed Prediction Time (sec)**	**CAD/CAM Error (%)**	**Proposed Prediction Error (%)**
Roughing	8000	10	1017.40	332	1032.80	-67.37	1.51
Finish Floor	8000	10	543.75	213	606.75	-60.83	11.59
Finish Walls	8000	10	133.31	92	114.30	-30.99	-14.26
Roughing	8000	20	752.40	332	711.05	-55.87	-5.50
Finish Floor	8000	20	435.01	213	420.85	-51.04	-3.26
Finish Walls	8000	20	125.61	92	111.01	-26.76	-11.62
Roughing	8000	50	551.52	332	478.66	-39.80	-13.21
Finish Floor	8000	50	334.37	213	368.28	-36.30	10.14
Finish Walls	8000	50	115.92	92	107.90	-20.63	-6.92

Table 2.4: Machining cycle time comparison for Aerostructure part case study.

			Total Machining Cycle Times			Errors (%)	
Case	**Feedrate (mm/min)**	**Tolerance (microns)**	**Measured (sec)**	**CAD/CAM (sec)**	**Proposed Prediction (sec)**	**CAD/CAM**	**Proposed Prediction**
1	8000	10	1694.46	637	1753.85	-62.41	3.50
2	8000	20	1313.02	637	1242.91	-51.49	-5.34
3	8000	50	1001.81	637	954.84	-36.42	-4.69

Table 2.5: Total machining cycle times and errors for measured, predicted and CAD/CAM.

2.4.3 Case Study 4 - Accurate Cutting Force Prediction using Predicted Feedrates

Lastly, the importance of accurate feedrate prediction for virtual machining models is demonstrated. This is realized by estimating cutting forces along the complex trochoidal toolpath shown in Fig.2.16. Predicting the cutting forces, considering the complex tool engagements on this toolpath, is realized by adapting the cutting force prediction model presented in [6] with the proposed feedrate prediction method. Readers should refer to [4] and [6] for details of the cutting force model.

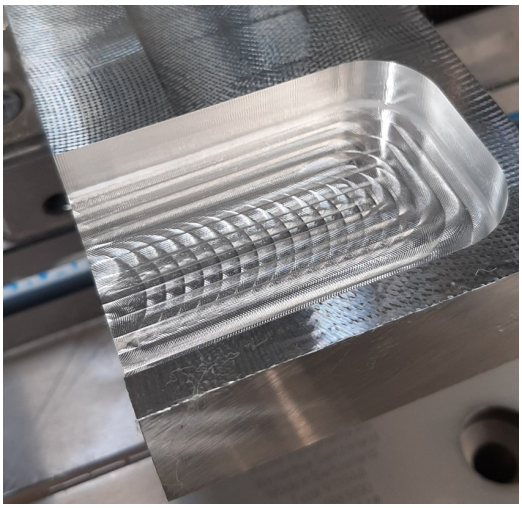

Figure 2.15: Machined AL7075 pocket using a trochoidal toolpath

To validate the feedrate prediction method with a cutting force model machining trials were conducted on the 5-axis DMG Mori eVo 40 machining centre fitted with a Heidenhain TNC640 controller. The toolpath, shown in Fig. 2.16, was designed using NX CAM as a trochoidal pocketing operation. A 40mm x 60mm x 10mm open sided pocket was selected as the test feature as shown in Fig. 2.15. A 2-fluted 12mm solid carbide end mill with a HSK-63A tool holder was used. The workpieces were 236mm x 30mm x 6mm aluminium 7075, each held using a Geradi compact grip vice mounted to the dynamometer. A Kistler 9139AA dynamometer and a National Instruments USB-6343 multi-channel DAQ was used to acquire cutting force data at 10kHz. The machining centre was connected to a local area network via a RJ45 network cable such that the machine controller data was accessed by two methods. The first using a pre-defined MTConnect datastream through a TCPIP connection at 20Hz and the second using an LSV2 protocol direct to the controller through a TCPIP connection at 111Hz.

The predicted cutting forces during the trochoidal section are shown in Fig. 2.16. The peak

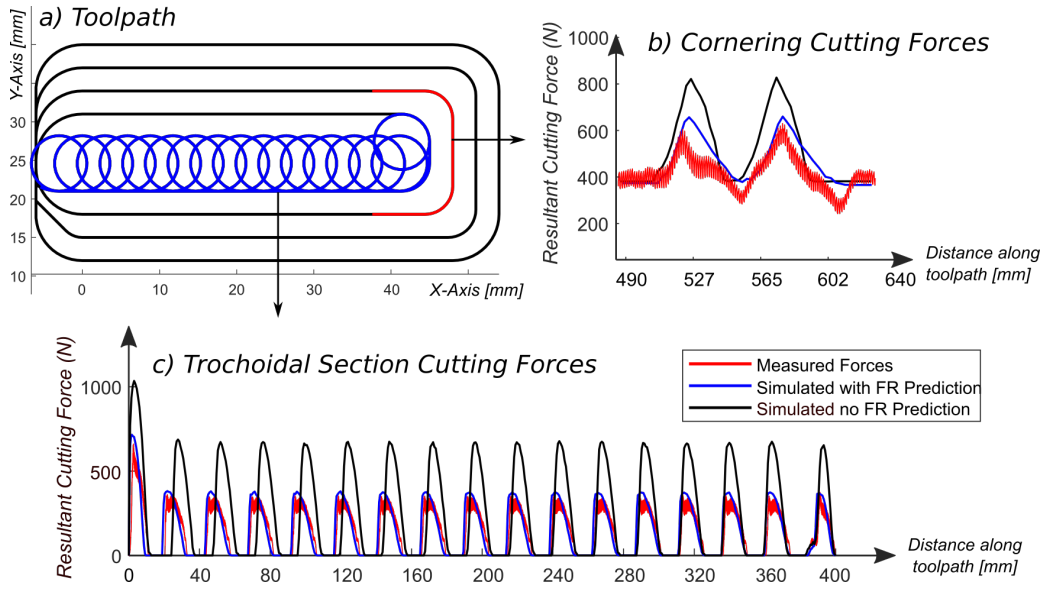

Figure 2.16: Simulated and measured resultant cutting forces for a trochoidal pocket

predicted cutting force for the standard feedrate model is 673N compared to 380N for the filtered feedrate model, from the peak measured cutting forces this gives prediction errors of 96.2% and 10.8% respectively. In the cornering section of the toolpath the peak predicted cutting force for the standard feedrate model is 821N compared to 656N for the filtered feedrate model, from the peak measured cutting forces this gives prediction errors of 37.3% and 9.7% respectively. The validation trials show that the inclusion of an accurate feedrate profile in the cutting force model enables a more accurate prediction of cutting forces for complex toolpaths.

2.5 Conclusions

A novel method of accurately modelling the trajectory generation of NC systems has been proposed. The main conclusions from this research are as follows:

1. An accurate method of feedrate prediction using FIR-based linear interpolation was introduced.

2. The method was demonstrated along short-segmented complex tool-paths.

3. The linear interpolation dynamics and commanded axis kinematic profiles of NC systems were predicted using both 2 and 3 first order Finite Impulse Response filters with the same time constant.

4. The corner blending behaviour during non-stop interpolation of linear segments was modeled by introducing velocity blending pulses.

5. For the first time, the minimum cornering feedrate, that satisfies both the tolerance and machining constraints, has been calculated analytically for toolpaths of any geometry.

6. The reduction in machining cycle time by using 3 FIR filters compared to 2 FIR filters was proven analytically.

7. The feedrate prediction method was validated experimentally against four different case studies demonstrating industrial 3-axis machining tool-paths.

8. The proposed method demonstrated cycle times can be estimated with >90% accuracy, greatly outperforming CAM-based predictions.

9. The predicted feedrate method was incorporated into a cutting force model, demonstrating an increase in cutting force accuracy for a complex toolpath, and validated experimentally.

Further work will integrate the methods into virtual machining and digital-twin models and extend the method to 5-axis machining. In addition to experimental trials, virtual NC kernels and NC emulators will be used to compare offline prediction methods.

Appendix 2A Kinematic Equations for 3-FIR Case

$$v'(t) = \begin{cases} \frac{F}{6T_1^3}t^3 & 0 \le t < T_1 \\ \frac{F}{2T_1^3}\left(-\frac{2}{3}t^3 + 3T_1t^2 - 3T_1^2t + T_1^3\right) & T_1 \le t < 2T_1 \\ \frac{F}{2T_1^3}\left(\frac{1}{3}t^3 - 3T_1t^2 + 9T_1^2t - 7T_1^3\right) & 2T_1 \le t < 3T_1 \\ F & 3T_1 \le t < T_v \\ \frac{F}{2T_1^3}\left(-\frac{1}{3}t^3 + T_vt^2 - T_vt + \frac{1}{3}T_v^3 + 2T_1^3\right) & T_v \le t < T_v + T_1 \\ \frac{F}{2T_1^3}(\frac{2}{3}t^3 - 2T_vt^2 - 3T_1t^2 + 3T_1^2t + 2T_v^2t + \ldots & \\ \ldots + 6T_vT_1t - \frac{2}{3}T_v^3 - 3T_v^2T_1 - 3T_vT_1^2 + T_1^3) & T_v + T_1 \le t < T_v + 2T_1 \\ \frac{F}{2T_1^3}(-\frac{1}{3}t^3 + T_vt^2 + 3T_1t^2 - T_v^2t - 9T_1^2t + \ldots & \\ \ldots - 6T_vT_1t + \frac{1}{3}\left(T_v + 3T_1\right)^3) & T_v + 2T_1 \le t < T_v + 3T_1 \end{cases} \tag{2.31}$$

$$a'(t) = \begin{cases} \frac{F}{2T_1^3}t^2 & 0 \le t < T_1 \\ \frac{F}{T_1^3}\left(-t^2 + 3T_1t - \frac{3}{2}T_1^2\right) & T_1 \le t < 2T_1 \\ \frac{F}{T_1^3}\left(\frac{1}{2}t^2 - 3T_1t + \frac{9}{2}T_1^2\right) & 2T_1 \le t < 3T_1 \\ 0 & 3T_1 \le t < T_v \\ \frac{F}{T_1^3}\left(-\frac{1}{2}t^2 + T_vt - \frac{1}{2}T_v^2\right) & T_v \le t < T_v + T_1 \\ \frac{F}{T_1^3}\left(t^2 - 2T_vt - 3T_1t + \ \frac{3}{2}T_1^2 + T_v^2 + 3T_vT_1\right) & T_v + T_1 \le t < T_v + 2T_1 \\ \frac{F}{2T_1^3}\left(-t^2 + 2T_vt + 6T_1t - T_v^2 - 9T_1^2 - 6T_vT_1\right) & T_v + 2T_1 \le t < T_v + 3T_1 \end{cases} \tag{2.32}$$

$$j'(t) = \begin{cases} \frac{F}{T_1^3}t & 0 \le t < T_1 \\ \frac{2F}{T_1^3}\left(-t + \frac{3}{2}T_1\right) & T_1 \le t < 2T_1 \\ \frac{F}{T_1^3}\left(t - 3T_1\right) & 2T_1 \le t < 3T_1 \\ 0 & 3T_1 \le t < T_v \\ \frac{F}{T_1^3}\left(-t + T_v\right) & T_v \le t < T_v + T_1 \\ \frac{2F}{T_1^3}\left(t - T_v - \frac{3}{2}T_1\right) & T_v + T_1 \le t < T_v + 2T_1 \\ \frac{F}{T_1^3}\left(-t + T_v + 3T_1\right) & T_v + 2T_1 \le t < T_v + 3T_1 \end{cases} \tag{2.33}$$

Appendix 2B Kinematic Profiles for 3-FIR Case

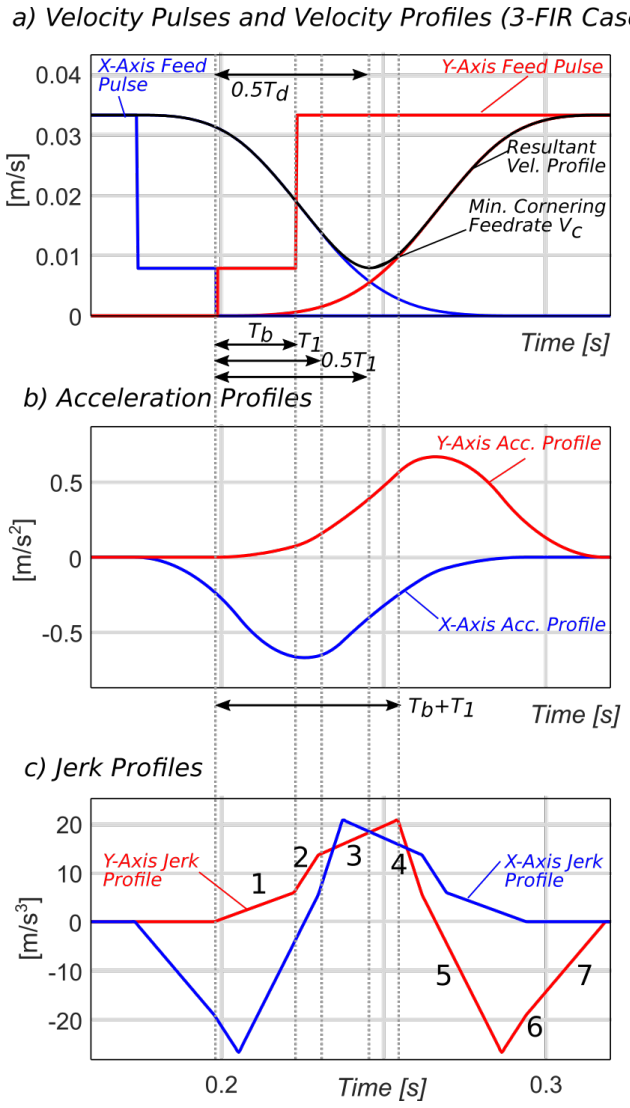

Figure 2.17: Velocity, acceleration and jerk profiles generated by blended velocity pulses interpolated 3-FIR filters

Appendix 2C Kinematic Equations for the 3-FIR Filter Case (Blending Pulses)

$$s'(t) = \begin{cases} \frac{\alpha F}{24T_1^3}t^4 & 0 \le t < T_b \\ \frac{2\alpha F}{T_1^3}\left(-\frac{1}{24}t^4 + \frac{1}{4}T_1t^3 - \frac{3}{8}T_1^2t^2\right) + \frac{\alpha F}{2}t - \frac{\alpha F{T_1}^2}{8} & T_b \le t < T \\ \frac{F\ (1-3\alpha)}{24T_1^3}\left(-t^4 + 4T_bt^3 - 6T_b^2t^2 + 4T_b^3t - T_b^4\right) + \dots & \\ + \dots \frac{\alpha F}{24T_1^3}(-8b_1t^3 + 18T_1^2 + 12T_b^2 - 36T_1T_b + 24T_b)t^2 + \dots & \\ \dots + \left(8T_b^2 - 12T_1^3\right)t + 3T_1^4 - 18T_b^4 + 24T_1T_b^3 - 16Tb^3b_1) & T_1 \le t < T_1 + T_b \end{cases} \tag{2.34}$$

$$v'(t) = \begin{cases} \frac{\alpha F}{6T_1^3}t^3 & 0 \le t < T_b \\ \frac{2\alpha F}{T_1^3}\left(-\frac{1}{6}t^3 + \frac{3}{4}T_1t^2 - \frac{3}{4}T_1^2t\right) + \frac{\alpha F}{2} & T_b \le t < T_1 \\ \frac{F(1-3\alpha)}{6T_1^3}(t^3 - 3T_bt^2 + 3T_b^2t + T_b^3) + \frac{\alpha F}{T_1^3}(\frac{1}{2}T_1^3 - \frac{1}{3}T_b^3 + \dots & \\ \dots - (\frac{3}{2}T_1^2 - 3T_1T_b + T_b^2)t + b_1t^2 - 2T_bb_1t) & T_1 \le t < T_1 + T_b \end{cases} \tag{2.35}$$

$$a'(t) = \begin{cases} \frac{\alpha F}{T_1^3}t & 0 \le t < T_b \\ \frac{2\alpha F}{T_1^3}\left(-\frac{1}{2}t^2 + \frac{3}{2}T_1t - \frac{3}{4}T_1^2\right) & T_b \le t < T_1 \\ \frac{2\alpha F}{T_1^3}\left(\frac{3}{2}T_1^2 - \frac{3}{2}T_1T_b + \frac{1}{2}T_b^2 - T_bb_1 + b_1\right) + \dots & \\ \dots - \frac{F(1-3\alpha)}{T_1^3}\left(\frac{1}{2}T_b^2 - \left(T_bt - \frac{1}{2}t^2\right)\right) & T_1 \le t < T_1 + T_b \end{cases} \tag{2.36}$$

$$j'(t) = \begin{cases} \frac{\alpha F}{T_1^3}t & 0 \le t < T_b \\ \frac{\alpha F}{T_1^3}\left(-t + \frac{3}{2}T_1\right) & T_b \le t < T_1 \\ \frac{F(1-3\alpha)}{T_1^3}\left(t - T_b\right) + \frac{2\alpha F}{T_1^3}\left(\frac{3}{2}T_1 - T_b\right) & T_1 \le t < T_1 + T_b \end{cases} \tag{2.37}$$

where $b_1 = \frac{3}{2}T_1 - T_b$.

The maximum TCP error occurs at $t = \frac{T_d}{2} = \frac{3}{2}T_1$ for the 3 first order FIR filter case, the interpolated axis velocity and displacement, are defined as equations (2.38) and (2.39) respectively:

$$v' = \frac{F}{48\,{T_1}^3}\left(36\,T_1T_b^2 - 54\,T_1^2\,T_b - 3\,T_1^3\,\alpha + 8\,T_b^3\,\alpha + 27\,T_1^3 - 8\,T_b^3 - 36\,T_1\,T_b^2\,\alpha + 54\,T_1^2\,T_b\,\alpha\right) \tag{2.38}$$

$$s' = -\frac{F}{384T_1^3}\left(96\,T_1\,T_b^3 + 216\,T_1^3\,T_b + 3\,T_1^4\,\alpha + 16\,T_b^4\,\alpha - 81\,T_1^4 - -16\,T_b^4 - 216\,T_1^2\,T_b^2 + \dots\right.$$
$$\left. \dots + 216\,T_1^2\,T_b^2\,\alpha - 96\,T_1\,T_b^3\,\alpha - 216\,T_1^3\,T_b\,\alpha\right) \tag{2.39}$$

Using equation (2.14), (2.38) and (2.39) can be expressed in terms of F and α as:

$$v' = \frac{F}{48}\left(-8\alpha^4 - 4\alpha^3 + 6\alpha^2 + 29\alpha + 1\right) \tag{2.40}$$

$$s' = \frac{F}{384}T_1\left(-16\alpha^5 - 16\alpha^4 + 8\alpha^3 + 16\alpha^2 + 85\alpha + 1\right) \tag{2.41}$$

Solving equation (2.27) with (2.39) results in the maximum TCP error for the 3-FIR filter case as follows:

$$\varepsilon_{TCP} = \frac{\sqrt{2}}{384}\sqrt{F^2\,{T_1}^2\,\left(\cos\theta_{TCP} + 1\right)\,\left(-16\,\alpha^5 - 16\,\alpha^4 + 8\,\alpha^3 + 16\,\alpha^2 + 85\,\alpha + 1\right)^2} \tag{2.42}$$

Chapter 3

Journal Paper 2 - Five-axis Trajectory Generation considering Synchronisation and Non-linear Interpolation Errors

3.1 Introduction

The fundamental advantage of 5-axis machine tools over their 3-axis counterparts is that they allow continuous interpolation of tool axis orientation synchronously with the tool centre point (TCP). To interpolate such 6 degree of freedom (DOF) tool motion, part programs (NC toolpaths) for 5-axis machining contain discrete position commands for all the 5 axes. This format is called the machine coordinate system (MCS) part programming and requires a machine specific computer-aided manufacturing (CAM) post-processor. Another approach is to directly provide the tool centre points (TCP) in Cartesian workpiece coordinate system and the associated unit tool orientation vectors in spherical coordinates. This direct part programming is called the WCS programming and does not require any post processor, which makes it general for any 5-axis machine tool. Therefore, WCS based part programming has become widely adopted for generating high speed 5-axis machining toolpaths. However, for numerical control (NC) systems of 5-axis machine tools, accurately and smoothly interpolating part programs defined in WCS requires overcoming series of challenges. This paper presents a novel approach for 5-axis NC systems to accurately and smoothly interpolate part programs in real-time.

Firstly, WCS based part programs use series of G01 moves, that command TCP to be interpolated linearly in the Cartesian workpiece coordinates and tool orientation vectors (ORIs) to be interpolated linearly in spherical coordinates. Interpolating TCPs linearly in Cartesian coordinates is not a challenging task. However, linear interpolation of ORIs synchronously with the TCP poses challenges. The simplest approach is to apply inverse kinematic transformations (IKT) at the start and end points of the linear move to compute corresponding axis positions. Discrete axis positions are then linearly interpolated in the MCS system, which is straightforward and most suitable for real-time programming. Figure 3.1 shows the relationship between MCS and WCS for a rotary table machine tool. Furthermore, IKT is applied only limited times, which makes it computationally inexpensive. Therefore, it has been well used in modern 5-axis NC systems. However, interpolating in MCS induces a nonlinear interpolation error which if not controlled can violate machining tolerances [32, 33] and as such, this research presents compensating control strategies.

Along a toolpath, the junctions between the G01 segments are discontinuous which can lead to machining problems as the motion must instantaneously stop at each respective junction. The discontinuity increases the overall machining cycle time, applies abrupt commands to the drives due the rapid deceleration and acceleration and ultimately results in poor surface finish. This issue has been well researched and many methods exists to overcome the discontinuous nature of linear G01 moves, which still remains the preferred method of commanding the machining of complex parts in industry.

The starting point for smoothing the corners between linear segments of a toolpath is by replacing them with parametric splines. Typically, this involves a 2-step method, first the corner is geometrically smoothed followed by local feedrate scheduling to meet the imposed kinematic constraints. Methods were developed to minimize chord errors and minimize feedrate fluctuations [34] during the 2-step process. Multiple approaches using polynomials of different order have been researched. Erkorkmaz et al.[19] inserted quintic splines to smooth out the sharp transitions in toolpaths which enabled corners to be transited whilst maintaining a continuous feedrate profile [35]. Sencer [36] fitted quintic B-splines to blend corner segments whilst satisfying axis acceleration limits and C^2 continuity by controlling the cornering error analytically. Ernesto and Fouriki inserted conic splices and used Bernstein form polynomials to solve the feedrate as a constrained linear programming problem [37], where the method was applied offline during pre-processing. A significant issue encountered with the polynomial methods is the

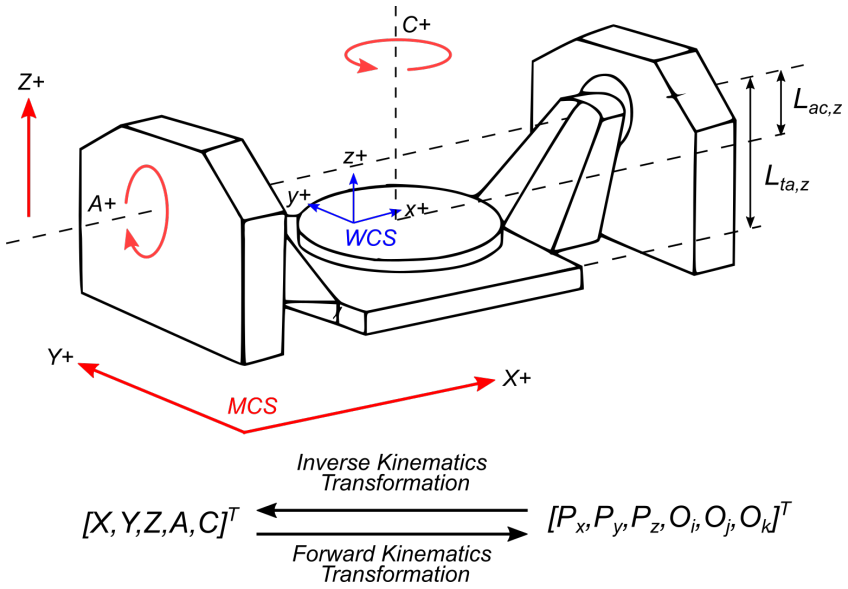

Figure 3.1: Five axis A/C configuration machine tool

computational bottleneck which limits some of the methods to offline use, leading to a focus on real-time interpolation [38, 39].

Extending geometric smoothing and feedrate scheduling to 5-axis adds complexity as the rotary and translational axes are smoothed separately yet must be synchronised. Focusing on 5-axis machining, Beudaert [40] introduced a corner rounding model to control the contour and orientation tolerances in the WCS. The TCP and ORI were each represented by separate B-spline curves. Using 5th and 6th order micro-splines for TCP and ORI respectively, Tulsyan and Altintas [41] calculated optimal control points using a newton Raphson method to achieve C^3 continuity at segment junctions whilst respecting tolerance limits. Focusing on spline curves Sun and Altintas [42] analytically calculated the control points to achieve C^3 continuous path smoothing, satisfying set tolerance whilst removing the requirement for iterative solutions. Huang et al. [43] used cubic B-splines for real-time C^2 continuous local corner and tool orientation smoothing in which synchronisation was achieved through sharing curve parameters. Bi et al. [44], interpolating in the MCS, used cubic Bezier curves to smooth the segment junctions of the translational and rotational paths. The method considered both TCP and tool orientation error using admissible approximation errors by relating the WCS tolerances to the MCS system using the Jacobian of the forward kinematics. Yang and Yuen [45] decoupled the interpolation by smoothing the TCP in the WCS and ORI in the MCS.

In 5-axis machining, the rotary axes also add an additional set of kinematic constraints to con-

sider during the local feedrate scheduling. Using cubic B-splines, Sencer at al [12], optimised feed along 5-Axis toolpaths by satisfying drive kinematic constraints. The method iteratively modulated the feed control points on the B-spline to ensure the commanded feed and drive limits were not exceeded. Beudaert et al. [14] developed an iterative method to compute the minimum time feedrate profile considering both translational and rotary kinematic limits.

More recently, there has been a growing field of research in using filtering methods for interpolation due to the computational efficiency. Tajima et al. [46] demonstrated accurate interpolation using FIR filters for 3-axis point-to-point (P2P) and continuous motion. Similar research by Li et al. [47] also reported the contouring error during corner transitions could be controlled by the introduction of overlaps and dwells within the velocity pulse train. Ward et al. [48] constrained TCP contouring error when conducting FIR-based interpolation by controlling cornering feedrate and simultaneously satisfying jerk limits for toolpath geometry. Tajima and Sencer [25] extended the FIR-based interpolation method to 5-axis toolpaths. Using the same FIR filter time constant, they synchronised tool motion, whilst blending cornering feedrates using blending velocity pulses. Both TCP and ORI motions were interpolated in the WCS. Decoupling the translational and rotary motion, Liu et al. [49] conducted FIR based interpolation in the WCS and MCS respectively. Their work considered both the axial and tangential kinematic constraints in their method. Jiang et al. [50] introduced a two-step interpolation process considering kinematic constraints of tool orientation error. The method smooths tool position orientation trajectories using cubic B-splines followed by FIR-based filtering. The tool position and orientation error for 5-axis interpolation was considered by Tajima et al. [51]. Through short velocity pulses smaller than the total filter delay, the pulses were formulated into an optimisation problem to minimize the contouring error during cornering transitions.

As cited, many papers have exploited the rotary axes in the MCS as a mechanism to smooth the orientation trajectory and minimize orientation errors in the WCS. Two main approaches have been taken. The first, based on geometry, blends the trajectory between the two rotary axes. Beudaert et al. [52] iteratively smoothed the rotary axes in MCS using splines under drive constraints. Similarly, Bi et al. [44] used cubic Bezier curves, Yang and Yuen [45] inserted quintic microsplines and finally Huang et al. [53] used biathoid curves. The second approach focuses on feedrate blending in the MCS where Liu et al. [49] used an FIR filtering approach and Zhang et al. [54] used a feedrate blending method under geometric and kinematic constraints. The referenced research all have two things in common. First, they all attempt to smooth adjoining

linear trajectories or motions in the MCS, and secondly they all evaluate the tool orientation errors at the corners or CL points. However, they do not consider the tool orientation error between successive tool orientation commands. For example, when the trajectory between two rotary positions is interpolated in the MCS, the resulting tool motion in the WCS does not follow the shortest distance path between the orientations. Hence, a significant nonlinear ORI interpolation error can occur at the midpoint of the tool motion. The orientation error is independent of the interpolation method and solely based on the properties of spherical geometry. Zhang et al. [33] presented a method to control this nonlinear error using a spherical angle linear interpolation and Geng et al. [32] generalised the error analysis to machine tools of any orthogonal configuration. We propose a new method to analytically calculate the nonlinear orientation error induced by interpolating in the MCS without the requirement for spherical interpolation. Furthermore, we present a strategy to control the nonlinear interpolation error.

In addition to addressing nonlinear interpolation of 5-axis machining, this paper presents a new method of decoupling the interpolation of the Cartesian and rotary axes. Within FIR-based interpolation, the acceleration and jerk limits can be guaranteed by selecting the appropriate FIR filter time constant. In order to synchronise the motion of the axes, the Cartesian FIR filter time constant is lowered to match the rotary axis FIR filter time constant. This reduces the available performance of the Cartesian axes by lowering the maximum acceleration and jerk limits; directly affecting the cornering feedrates during complex toolpaths and resulting in slower machining cycle times. It is therefore, beneficial to exploit the performance of both the Cartesian and rotary axes separately and decouple the FIR filter time constants. Furthermore, as demonstrated in this research, the magnitude of the Cartesian FIR filter time constant with respect to the rotary FIR filter time constant has a measurable effect on the tool orientation error and synchronisation. This paper presents a novel method of 5-axis interpolation using FIR filtering of the Cartesian and rotary axes which analytically constrains the TCP and ORI errors whilst satisfying both the Cartesian and rotary kinematic limits. The proposed method enables a faster Cartesian FIR filter time constant to be selected based on an analytically defined measure of synchronisation which improves the performance of the machine tool by reducing the overall machining cycle time.

3.2 Point-to-point (P2P) Linear Interpolation

In this section, firstly low-pass filtering based interpolation of P2P 5-axis machining toolpaths is presented. This is then followed by non-stop interpolation of contouring paths for high-speed machining.

3.2.1 Linear Interpolation of TCP

A basic P2P linear (G01) move commands the tool pose to travel linearly from the starting tool centre point $\mathbf{P}_s = [P_{s,x}, P_{s,y}, P_{s,z}]^T$ and tool orientation vector $\mathbf{O}_s = [O_{s,i}, O_{s,j}, O_{s,k}]^T$ to the final tool pose defined by $\mathbf{P}_e$ and $\mathbf{O}_e$. In 5-axis machining the machining feedrate F is typically defined as the linear speed of the tool center point (TCP), and such a linear move can be represented by a single feed pulse $v(t)_{TCP}$ (see Fig.3.2a):

$$v_{TCP}(t) = \begin{cases} F, & 0 < t \leq T_v \\ 0, & t > T_v \end{cases} \tag{3.1}$$

where $L = \|\mathbf{P}_e - \mathbf{P}_s\|_2$ and the duration of the feed pulse becomes $T_v = L/F$. Next, the tangential feed pulse $v_{TCP}(t)$ is resolved into its Cartesian TCP velocity pulse components:

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} = v_{TCP}(t)\mathbf{u}, \text{ where } \mathbf{u} = \frac{\mathbf{P_e} - \mathbf{P_s}}{L}, \tag{3.2}$$

and the TCP is interpolated linearly in the WCS by simply filtering the Cartesian velocity pulses by a low-pass FIR filter as:

$$\begin{bmatrix} v'_x(t) \\ v'_y(t) \\ v'_z(t) \end{bmatrix} = v_{TCP}(t)\mathbf{u} * m_{TCP}(t), \tag{3.3}$$

where * is the convolution operation, m_{TCP} is the impulse response of the FIR filter used for smoothing TCP velocity pulses, and $[v'_x, v'_y, v'_z]$ are the filtered (interpolated) commands. The low-pass FIR filter is defined as the following 3rd-order transfer function:

$$G_{TCP}(s) = \frac{\left(1 - e^{-sT}_{TCP,1}\right)\left(1 - e^{-sT}_{TCP,2}\right)\left(1 - e^{-sT}_{TCP,3}\right)}{s^3}, \tag{3.4}$$

where $T_{TCP,1}$,$T_{TCP,2}$ and $T_{TCP,3}$ are the time constants of the filter and s is the Laplace operator. Time constants control the frequency spectrum of the FIR filter and determine the

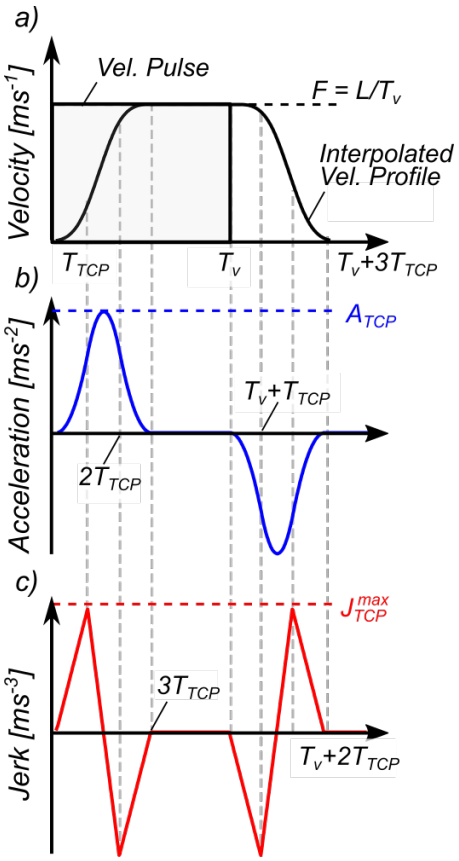

Figure 3.2: Kinematic profiles for an interpolated velocity pulse with 3 identical FIR filters

acceleration, peak acceleration and jerk values of the interpolated motion profile [48, 25, 21]. For Cartesian machine tools, FIR filter time constants can be selected carefully to avoid exciting lightly damped structural resonances of the machine tool while at the same time keeping the peak acceleration within the limit of the drives. However, due to the kinematic structure of 5-axis machine tools, time constants of the FIR filter cannot be assigned to precisely target resonances in Cartesian directions of the machine. This is due to the fact that the combination of Cartesian and rotary axes control the TCP motion and the direction of the acceleration varies constantly with the rotary axis positions. Therefore, instead of directly targeting structural resonances in each Cartesian directions of the machine tool, filter time constants are tuned to control the peak value of acceleration and jerk of the TCP to indirectly impose drive acceleration limits while at the same time avoiding residual vibrations.

The use of 3 matching FIR filters for interpolating (smoothing) the TCP motion was introduced in [48], where $T_{TCP} = T_{TCP,1} = T_{TCP,2} = T_{TCP,3}$. As a result, once convolved with a pulse velocity command, the time derivatives of the resulting interpolated velocity yield a bell-shaped acceleration profile and a triangular jerk profile as shown in Fig.3.2. Derived in [48], T_{TCP} can

be selected to satisfy both the acceleration and jerk limits of the TCP motion in WCS as:

$$T_{TCP} = \max\left\{\frac{3\Delta F}{4A_{TCP}^{\max}}, \sqrt{\frac{\Delta F}{J_{TCP}^{\max}}}\right\}. \tag{3.5}$$

Finally, filtered TCP velocity pulses are integrated to generate the final TCP position commands:

$$\begin{bmatrix} x'(t) \\ y'(t) \\ z'(t) \end{bmatrix} = \int_0^t \begin{bmatrix} v'_x(t) \\ v'_y(t) \\ v'_z(t) \end{bmatrix} d\tau. \tag{3.6}$$

The interpolated displacement profiles can be expressed in piecewise analytical form (see Appendix A in [48]) , which are used extensively within the scope of this research.

3.2.2 Linear Interpolation of Tool Axis Orientation

The interpolation of TCP in the WCS is presented above. The tool orientation, on the other hand, is interpolated in the MCS. The strategy is depicted in Fig.3.3. Inverse kinematics transformation for the rotary drives are applied to the starting and ending tool orientation vectors, $\boldsymbol{O_s}$ and $\boldsymbol{O_e}$ of the G01 command, and rotary axis positions are computed as

$$\begin{bmatrix} A_s \\ C_s \end{bmatrix} = IKT_{ROT}\left(O_{i,s}, O_{j,s}, O_{k,s}\right), \tag{3.7a}$$

$$\begin{bmatrix} A_e \\ aC_e \end{bmatrix} = IKT_{ROT}\left(O_{i,e}, O_{j,e}, O_{k,e}\right), \tag{3.7b}$$

where IKT_{ROT} is the inverse kinematics transformation for calculating the rotary axis positions (A,C) only. For a generic 5-axis machine tool with an A/C rotary table configuration (Fig.3.1), the inverse kinematics for the rotary axes becomes [55]:

$$IKT_{ROT}\begin{cases} A = -\sin^{-1}\left(O_k\right), & (-\pi < A \leq \pi) \\ C = \tan^{-1}\left(O_i, O_j\right). & (O_i = O_j \neq 0) \end{cases} \tag{3.8}$$

Notice that IKT_{ROT} does not require the TCP Cartesian coordinates. The rotary axis motion is then interpolated by generating its pulse velocity commands in the MCS, refer to Fig.3.3b, as:

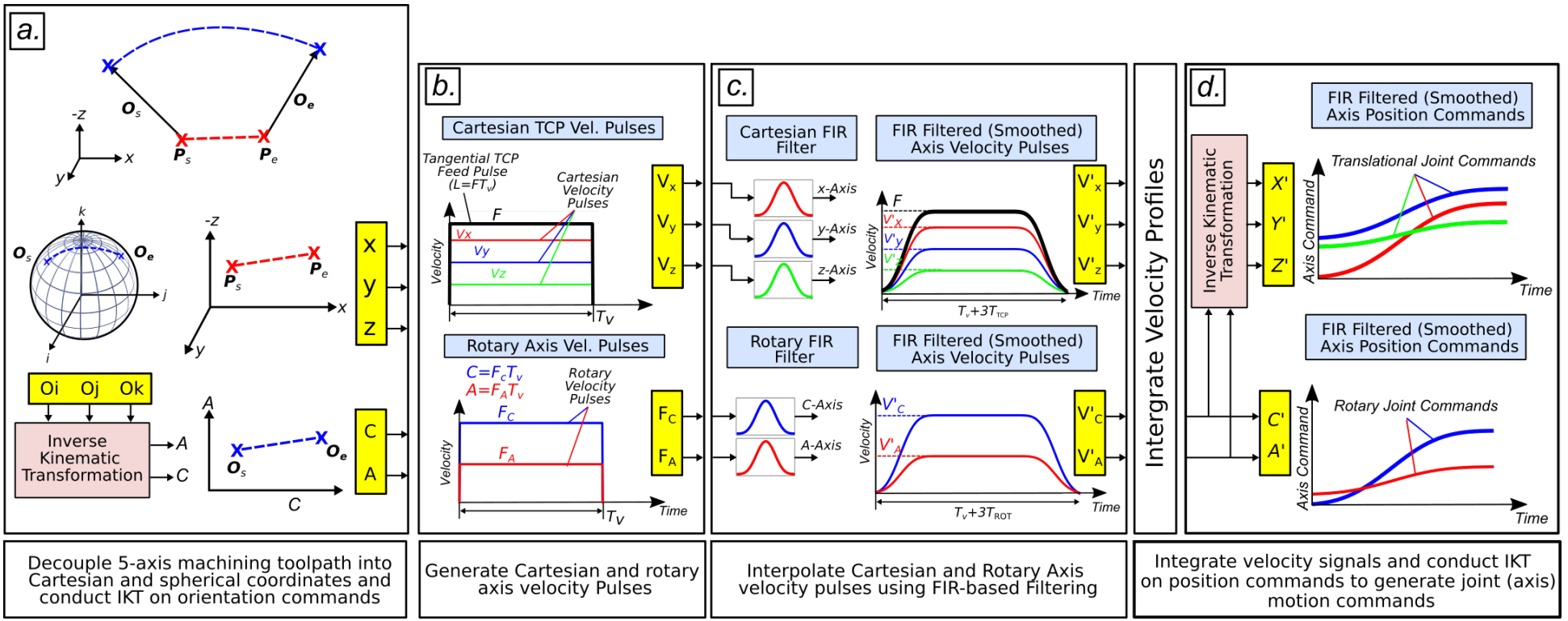

Figure 3.3: Joint WCS-MCS Linear Interpolation of P2P 5-Axis Motion

$$\begin{bmatrix} F_A(t) \\ F_C(t) \end{bmatrix} = \begin{bmatrix} A_e - A_s \\ C_e - C_s \end{bmatrix} / T_v, \quad 0 < t \leq T_v \tag{3.9}$$

and low-pass filtering them by a separate FIR filter, $G_{ROT}(s)$ as shown in Fig.3.3c. The filter $G_{ROT}(s)$ is designed with the same structure and order as the filter $G_{TCP}(s)$ given in Eq.(3.4), but its time constant T_{ROT} can be selected differently from that of G_{TCP} . As it will be described in the following sections, setting T_{ROT}:

$$T_{ROT} = T_{ROT,1} = T_{ROT,2} = T_{ROT,3} > T_{TCP}, \tag{3.10}$$

leads to better utilization of the rotary kinematic limits. Specifically, such assignment is advantageous since torque and jerk capacity of rotary drives used in typical 5-axis machine tools are much smaller than the ones used for Cartesian drives. The advantages, such as attaining faster cycle times in non-stop interpolation of complex 5-axis toolpaths, will be clarified in the following sections.

The rotary axis velocity pulses are then filtered by $G_{ROT}(s)$:

$$v'_{ROT}(t) = \begin{bmatrix} v'_A(t) \\ v'_C(t) \end{bmatrix} = \begin{bmatrix} F_A(t) \\ F_c(t) \end{bmatrix} * m_{ROT}(t), \tag{3.11}$$

where $m_{ROT}(t)$ is the impulse response of $G_{ROT}(s)$, and $v'_{ROT}(t)$ is integrated to compute the final rotary axis positions as shown in Fig.3.3d,

$$\begin{bmatrix} A'(t) \\ C'(t) \end{bmatrix} = \int_0^t \begin{bmatrix} v'_A(t) \\ v'_c(t) \end{bmatrix} d\tau. \tag{3.12}$$

Finally, the MCS Cartesian axis positions $[X', Y', Z']$ are generated by applying the inverse kinematics for the TCP:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = IKT_{TCP}\left(x', y', z', A', C'\right), \tag{3.13}$$

where IKT_{TCP} is the TCP's inverse kinematics, which requires both the interpolated TCP positions (x', y', z') and the rotary axes (A', C'). For a typical 5-axis machine tool with a A/C type rotary axes the Cartesian inverse kinematics can be expressed as [41]:

$$IKT_{TCP} = \begin{cases} X = -x\cos C + y\sin C \\ Y = -x\cos A\sin C - y\cos A\sin C + z\sin A + L_{ac,z}\sin A \\ Z = x\sin A\sin C + y\sin A\cos C + z\cos A + L_{acz}, \cos A - L_{ta,z} \end{cases}, \quad (3.14)$$

where $L_{ac,z}$ and $L_{ta,z}$ are linear offsets between the rotary drive coordinate frames (see Fig.3.1). Notice that in the proposed approach since the TCP is accurately interpolated in the WCS, it does not show any nonlinear interpolation errors along P2P toolpaths. However, the tool orientation is interpolated in the MCS, and thus tool axis orientation is not interpolated linearly in the spherical coordinates. It exhibits interpolation errors [33, 32]. The following section models such interpolation errors and proposes a discretization method to constrain them along long linear toolpaths.

Formulation of nonlinear interpolation errors

The trajectory of the tool axis orientation differs based on where it is interpolated; namely, in WCS or in the MCS. Fig.3.4 depicts the path of the tool axis orientation when it is interpolated in the WCS using spherical linear interpolation [25], and when it is linearly interpolated in MCS using the rotary axes. When interpolated in WCS on the unit sphere using either spherical linear interpolation [25] or quaternions [56], the angular path of the tool axis follows the shortest distance between the starting and ending ORI vectors, $\boldsymbol{O_s}$ and $\boldsymbol{O_e}$. However, when interpolated linearly in the MCS by the rotary drives, tool axis orientation follows a curved path. This is due to the fact that spherical interpolation rotates the tool axis around a fixed vector defined by $\boldsymbol{O} = \boldsymbol{O_s} \times \boldsymbol{O_e}$. Whereas, due to the position dependent rotational inverse kinematic transformation, tool axis orientation makes a curved path on the unit sphere as shown in Fig.3.4b. Such deviation depends on the rotary axis position and may become significantly large for long G01 moves. In this research, we consider WCS based linear interpolation as ground truth, and the deviation, the nonlinear MCS interpolation error ε_{MCS}, is predicted analytically. An adaptive path discretization method is also developed to confine such interpolation errors under a user-specified tolerance value $\varepsilon_{ORI,Tol}$.

Firstly, the angular error (deviation) due to MCS interpolation is modelled using spherical geometry based on rotary axis position commands. Fig.3.4b shows detailed angular tool trajectories based on WCS and MCS interpolation, and Fig.3.4c shows the annotated trajectories depicted as spherical triangles. The points C and B denote the $\boldsymbol{O_s}$ and $\boldsymbol{O_e}$ vectors on unit

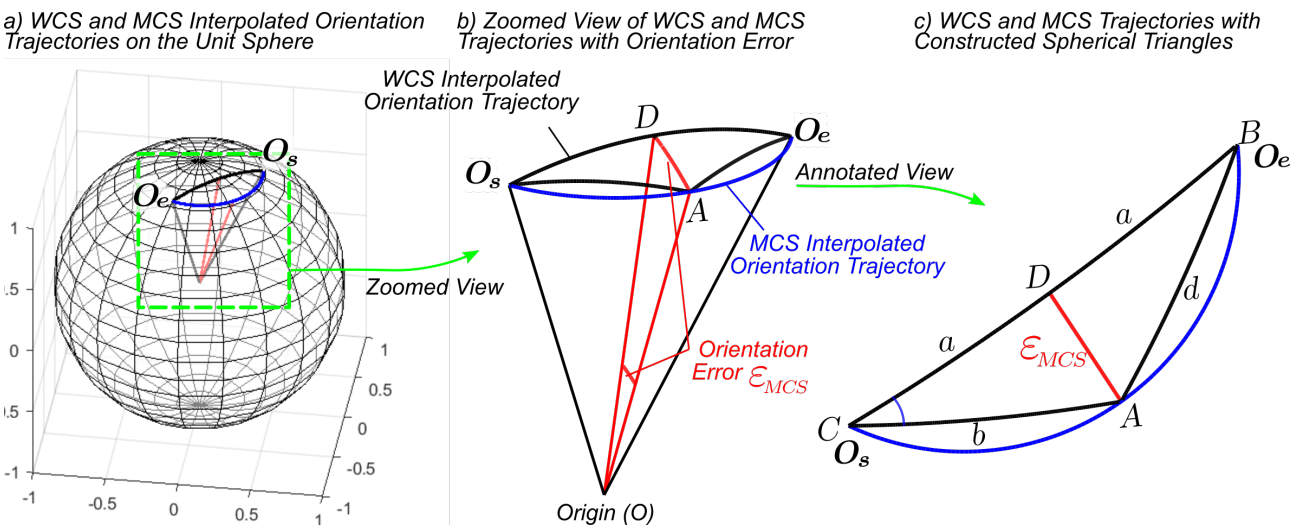

Figure 3.4: WCS and MCS interpolated orientation trajectories, (b) zoomed view, (c) annotated view

sphere, respectively. Maximum deviation from the WCS interpolated path occurs right in the middle of the linear move as the tool travels from C to B at point mid-point D. Appendix 3A analytically proves that the maximum deviation occurs at this bisection of the trajectory. The spherical triangles in Fig.3.4c are trajectories generated on the unit sphere. Therefore, the arc lengths between marked points also correspond to angles. The point D divides the trajectory into 2 equal arc sections with length a. The length of the arc ε_{MCS} depicts the maximum angular deviation between WCS and MCS interpolated orientation trajectories. Notice that since MCS interpolated path is curved, point A divides the MCS interpolated path into 2 unequal arc sections with lengths of b and d where ABC and ADC form spherical triangles. To calculate the great circle arc length ε_{MCS}, the haversine formula [57] is adapted as follows. An arc length θ on a great circle trajectory connecting two points (shown in Fig.3.5a) can be calculated by

$$\theta = 2r \arcsin(\sqrt{h}), \tag{3.15}$$

where

$$h = \frac{1 - \cos(\varphi_2 - \varphi_1)}{2} + \cos(\varphi_1)\cos(\varphi_2)\frac{1 - \cos(\lambda_2 - \lambda_1)}{2}, \tag{3.16}$$

and λ and φ represent the latitude and longitude of those points and r is the radius of the sphere. As noted from Fig.3.5b, the C-axis rotates the unit sphere and directly controls λ by $C = \lambda$, whereas the A-axis controls the longitude φ by $A = \frac{\pi}{2} - \varphi$. Thus, Eq.(3.16) can be modified with rotary axis positions as:

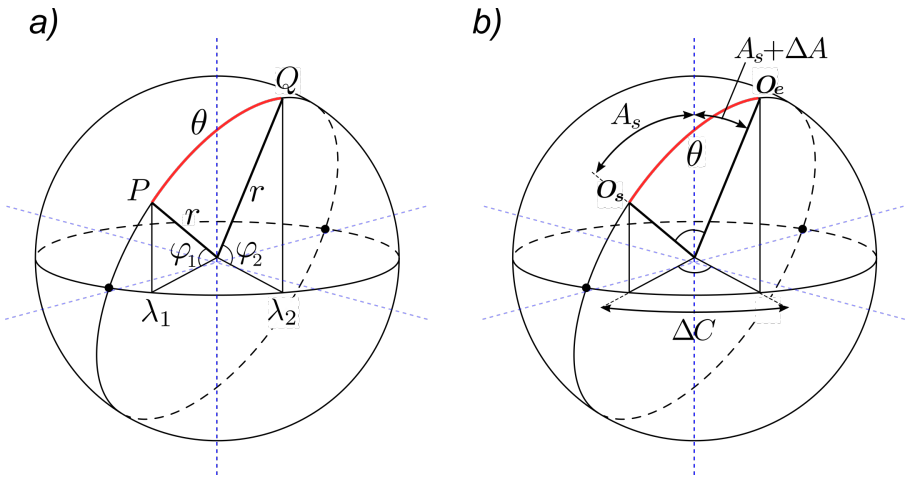

Figure 3.5: Great-circle distance (drawn in red) between two points on a sphere, P and Q, (b) great circle arc length on the unit circle using rotary axis variables.

$$h = \frac{1-\cos\left(-A_s+A_e\right)}{2} + \cos\left(\frac{\pi}{2}-A_s\right)\cos\left(\frac{\pi}{2}-A_e\right)\frac{1-\cos\left(-C_s+C_e\right)}{2}. \tag{3.17}$$

Implementing the following basic trigonometric identities $\cos\left(\pi/2\ -\theta\right) = \sin\theta$ and $\sin\left(A_e\right) = \sin\left(A_s+\Delta A\right)$, Eq.(3.17) becomes:

$$h = \frac{1-\cos(-\Delta A)}{2} + \sin\left(A_s\right)\left(\sin\left(A_s\right)\cos(-\Delta A) - \cos\left(A_s\right)\sin(-\Delta A)\right)\frac{1-\cos(\Delta C)}{2}, \tag{3.18}$$

where $\Delta C = C_e - C_s$ and $\Delta A = A_e - A_s$. Setting $r = 1$, Eq.(3.18), can be used to compute unknown arc lengths from $a = 1\arcsin(\sqrt{h_{BC}})$, $b = 2\arcsin(\sqrt{h_{AC}})$ and $d = 2\arcsin(\sqrt{h_{AB}})$. The full forms of h_{BC}, h_{AC} and h_{AB} are included in Appendix 3B. Next, the spherical law of cosines is applied to calculate the angle $\angle C$ from Fig. 3.4c,

$$\angle C = \cos^{-1}\left(\frac{\cos d - \cos 2a\cos b}{\sin 2a\sin b}\right), \tag{3.19}$$

and finally, the arc length ε_{MCS} can be computed using the inner spherical triangle ACD using:

$$\varepsilon_{MCS} = \cos^{-1}\left(\cos a\cos b + \sin a\sin b\cos\angle C\right), \tag{3.20}$$

which represents the maximum angular deviation between the WCS and MCS interpolated paths. Substituting the expressions for a, b, ε_{MCS} and $\angle C$ into Eq.(3.19) results in the orientation error due to spherical geometry as a function of initial A-axis position (A_s), A-Axis displacement (ΔA) and C-axis displacement (ΔC):

$$\varepsilon_{MCS} = \cos^{-1}\left(\frac{\cos\left(\frac{\Delta A}{2}\right)\left(\cos\left(\frac{\Delta C}{2}\right)+\cos(2A_s)\cos(\Delta A)-\sin(2A_s)\sin(\Delta A)-\cos\left(\frac{\Delta C}{2}\right)\cos(2A_s)\cos(\Delta A)+\cos\left(\frac{\Delta C}{2}\right)\sin(2A_s)\sin(\Delta A)+1\right)}{\sqrt{\cos(\Delta A)+\cos(2A_s)\cos(\Delta A)-\sin(2A_s)\sin(\Delta A)+\cos(\Delta A)\cos(\Delta C)-\cos(2A_s)\cos(\Delta A)\cos(\Delta C)+\sin(2A_s)\cos(\Delta C)\sin(\Delta A)+2}}\right) \tag{3.21}$$

Adaptive discretization for long toolpaths

An adaptive discretization algorithm for long tool-paths is developed to ensure that the MCS interpolation errors are kept within user defined tolerance. The idea is to divide a long toolpath into shorter segments to confine the MCS interpolation induced tool orientation error, and it is illustrated in Fig.3.6.

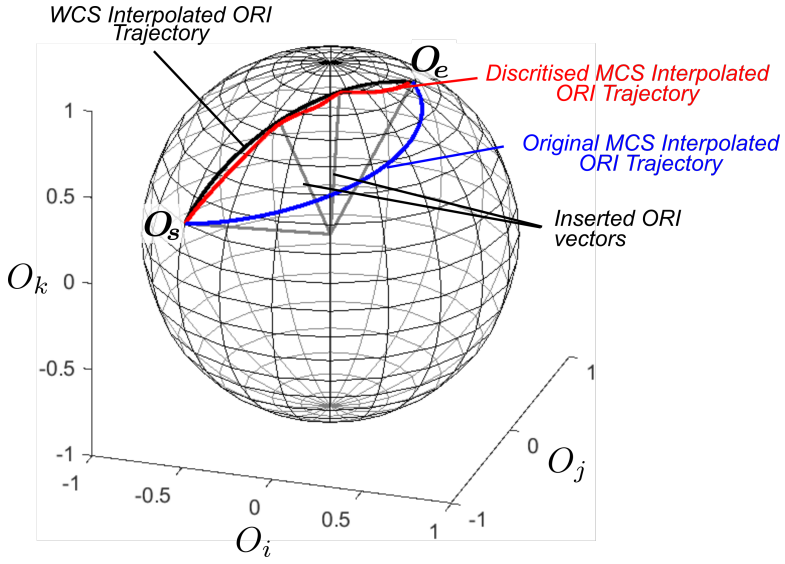

Figure 3.6: Original and discretized MCS interpolated ORI trajectories

As shown, a long toolpath is discretized into smaller shorter segments, and the angular interpolation errors are kept within a tolerance value by adding intermediate moves. Figure 3.7 shows the proposed strategy.

The strategy starts with calculating the starting and ending rotary axis positions $[A_s, C_s]$, $[A_e, C_e]$ from the rotary inverse kinematics, IKT_{ROT}. The tool orientation is determined from the initial bearing along the arc by,

$$\gamma = \tan^{-1}(m/n), \tag{3.22}$$

where

$$\begin{aligned} m &= \sin(\Delta C)\sin\left(A_s + \Delta A\right), \\ n &= \sin\left(A_s\right)\cos\left(A_s + \Delta A\right) - \cos\left(A_s\right)\sin\left(A_s + \Delta A\right)\cos(\Delta C). \end{aligned} \tag{3.23}$$

The parameterised rotary positions are calculated using $\gamma, A_s, \Delta A, \Delta C$ and are synchronised across the arc using the synchronisation parameter u where $u \in [0, 1]$,

$$A(u) = \cos^{-1}\left(\cos\left(A_s\right)\cos(\theta(1-u)) + \sin\left(A_s\right)\sin(\theta(1-u))\cos(\gamma)\right), \tag{3.24}$$

$$C(u) = C_s + \tan^{-1}(f/g), \tag{3.25}$$

where

$$\begin{aligned} f &= \sin(\gamma)\sin(\theta(1-u))\sin(A_s), \\ g &= \cos(\theta(1-u)) - \cos(A_s)\cos(A(u)), \end{aligned} \tag{3.26}$$

and θ is the arc length between the initial and final tool orientations calculated using the modified haversine formula (3.15) and (3.18). The nonlinear interpolation error along the arc length is then calculated using (3.21) for each incremental A and C axis position calculated in (3.24) and (3.25). A new orientation vector is inserted when $\varepsilon_{MCS} = \varepsilon_{ORI,Tol}$. The process repeats with a new initial bearing and starting rotary positions $[A_s{}^*, C_s{}^*]$. The process is terminated when $[A(u), C(u)] = [A_e, C_e]$. The result is a discretized toolpath which irrespective of the MCS-based interpolation method (i.e. geometric or velocity smoothing) will satisfy the ORI tolerance in between corner segments. Later sections will address the control of interpolation errors between successive CL points.

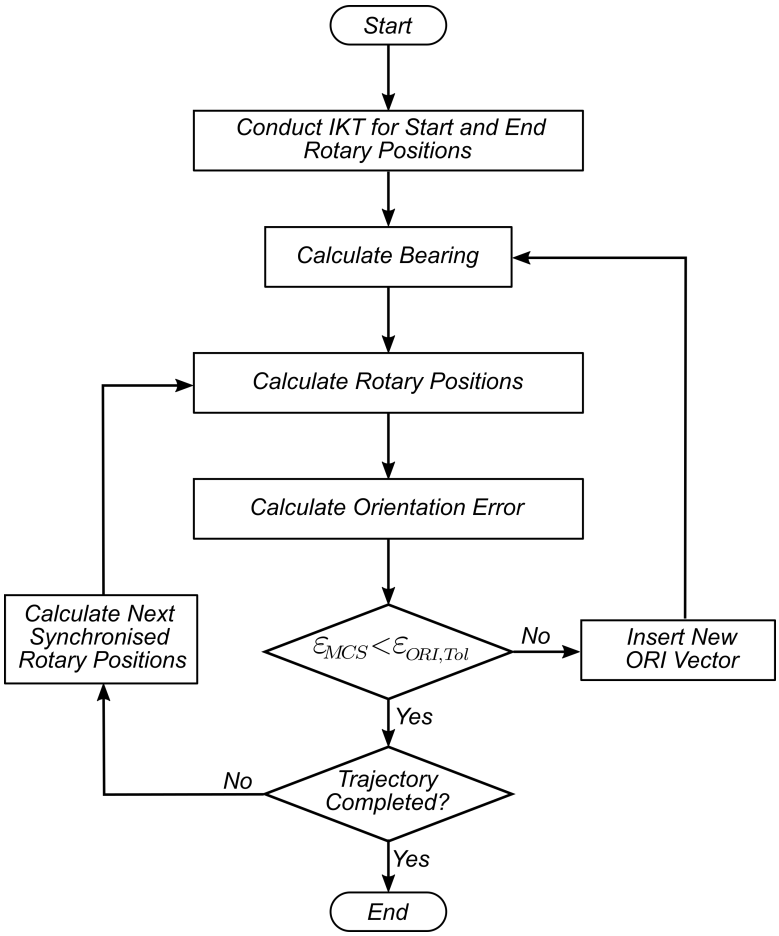

Figure 3.7: Flowchart showing method of controlling orientation error due to MCS interpolation

Control of Synchronisation Errors

Synchronization between the TCP and ORI may affect the surface finish [58]. In previous spline based research [59][60], synchronisation between the TCP and ORI motion was considered at segment junctions to ensure no abrupt changes in tool orientation due to the interpolation process. Poor synchronisation would result in acceleration or jerk spikes, thereby damaging the surface finish [42]. Therefore, the objective of this section is to characterize tool-pose synchronization errors and provide guidance in selecting time constants of rotary drives to attain both accurate and high-speed 5-axis motion.

The synchronization error during linear interpolation can be defined by the ratio of linear TCP's velocity to the angular velocity of the tool axis by:

$$\lambda(t) = \frac{v_{TCP}(t)}{v_{ORI}(t)}, \tag{3.27}$$

where v_{ORI} is tool axis' angular velocity. In physical terms, $\lambda(t)$ is equal to the ratio between the change in TCP displacement to the change in tool orientation. It is used as measure to address the different kinematic properties of the tool TCP and angular motions. For best synchronization the linear/angular velocity ratio λ should be kept constant. Since tool motion is only time synchronized along a single CL-line, λ fluctuates along acceleration transients. This fluctuation is mainly controlled by the FIR filter time constants used for interpolating TCP and ORI. Firstly, as shown in Fig.3.8a, if time constants of the TCP and ROT FIR filters are matched, i.e. $T_{ROT} = T_{TCP}^{J_{max}}$, λ is smooth, and its extremum occurs around the acceleration/deceleration transients of the motion. If the rotary axis time constants T_{ROT} is selected identical as the T_{TCP}, rotary axes torque and jerk limits may be easily violated as shown in the third row of Fig.3.8a. This is due the fact that Cartesian axes typically are more "dynamic" providing larger acceleration and jerk limits as compared to the rotary axes. One way to satisfy rotary axis limits is to lower the feedrate as shown in Fig.3.8b, which significantly elongates overall cycle time deteriorating productivity. Another option is to set rotary axis time constants large and match TCP time constant to the rotary. Notice that such arrangement ensures that rotary drive limits are respected as illustrated in Fig.3.8c, but moderately increases overall cycle time.

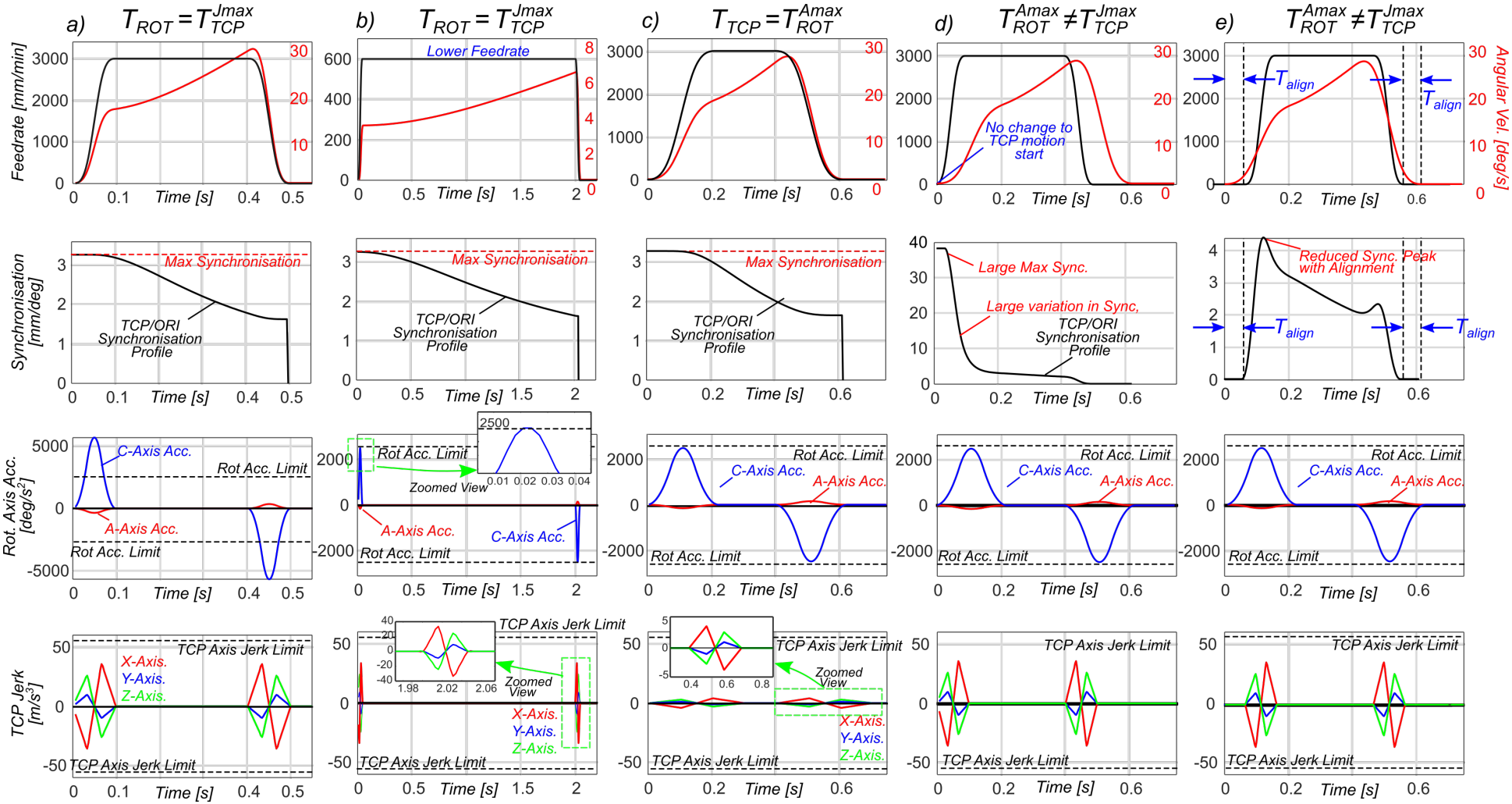

Figure 3.8: TCP and ORI Synchronisation - illustrating the effect of time constants on λ for multiple cases

Another way to improve productivity is to select larger time constants for the rotary axes $T_{ROT} > T_{TCP}$ so that both Cartesian and rotary axes limits are well utilised. Fig.3.8d shows the impact of mismatching TCP and ROT time constants. Firstly, rotary axes kinematic limits are respected without altering the TCP profiles. Please note that the cycle time on a simple P2P motion is not changed from the previous case where both TCP and ROT time constants are lowered. However, as it will be shown in the following sections, such arrangement will make sure that during high-speed contouring motion mismatched time constants would help utilise drive capacities more efficiently and even minimize overall cycle times. A key disadvantage is that the synchronization between TCP and ROT is deteriorated as indicated by largely fluctuating λ (See Fig.3.8d second row).

To synchronize the tool-pose motion, the start of the motion for the TCP velocity profiles is shifted by a small amount. In other words, slower rotary drives start their motion earlier and TCP motion is initiated afterwards. This is achieved by delaying the TCP velocity pulses by an amount of

$$T_{\text{align}} = \frac{3}{2}\left(T_{ROT} - T_{TCP}\right), \tag{3.28}$$

so that the TCP and ORI motion are better aligned. To maintain the static TCP position during the delay, the motion of the Cartesian feed drives in the MCS compensate to ensure the TCP is fixed and there is no impact to TCP error. The alignment T_{align} can be viewed as the half the difference between the two filter delays where $T_{delay,ROT} = 3T_{ROT}$ and $T_{delay,TCP} = 3T_{TCP}$. Fig.3.8e shows the aligned pulse profiles and its effect on the synchronization. As shown λ is more steady and lowered while axis kinematic limits are better utilised. However, notice that there are still peaks in λ, and as reported by literature [58], these are associated with a deteriorated surface finish. The synchronization error peaks can be characterized analytically as follows. Angular tool velocity v_{ROT} can be expressed from rotary axis velocities,

$$v_{ROT}(t) = \left\|\tfrac{d\mathbf{O}'}{dt}\right\|_2 = \|\mathbf{J}_{ROT}\mathbf{v}'_{ROT}\|_2 = \| \underbrace{\begin{bmatrix} \cos A \sin C & \cos C \sin A \\ -\cos A \cos C & \sin A \sin C \\ -\sin A & 0 \end{bmatrix}}_{\mathbf{J}_{ROT}} \begin{bmatrix} v'_A(t) \\ v'_c(t) \end{bmatrix} \|_2 = \sqrt{v'^2_A(t) + v'^2_C(t) \cdot \sin^2 A(t)}, \tag{3.29}$$

where $\boldsymbol{J}_{ROT} = \frac{d\boldsymbol{O}'}{d\theta}$ for $\theta = A, C$. λ can be derived as:

$$\lambda_{P2P}(t) = \frac{v_{TCP}(t)}{v_{ROT}(t)} = \frac{\sqrt{v'^2_x(t) + v'^2_y(t) + v'^2_z(t)}}{\sqrt{v'^2_A(t) + v'^2_c(t) \cdot \sin^2 A(t)}}. \tag{3.30}$$

The extremum of λ during non-stop contouring is difficult to determined analytically. However, for a P2P motion this occurs when $t = 0.5\,(3T_{ROT} + T_1)$, and the final expression for $|\lambda|_\infty$ becomes:

$$|\lambda|_\infty = \frac{20\sqrt{{F_x}^2 + {F_y}^2 + {F_z}^2}}{\sqrt{\frac{\left({F_A}^2 + {F_C}^2 \sin\left(\frac{F_A(-{T_{TCP}}^4 + 18{T_{TCP}}^2{T_{ROT}}^2 + 48T_{TCP}{T_{ROT}}^3 + 39{T_{ROT}}^4)}{192{T_{ROT}}^3}\right)^2\right)\left(-{T_{TCP}}^3 + 9T_{TCP}{T_{ROT}}^2 + 12{T_{ROT}}^3\right)^2}{{T_{ROT}}^6}}}. \tag{3.31}$$

Detailed derivation of Eq. (3.31) can be found in Appendix 3D. and it can be used to determine an upper limit on the time constant mismatch between the rotary and TCP. For instance, let us assume that the rotary axis FIR time constants are first set to respect their rotary axis kinematics, e.g. acceleration and jerk limits. Based on this, $T_{TCP} = nT_{ROT}$, $n \in [0, 1]$ is assigned, and the resultant synchronization and peak jerk value of TCP can be related as shown in Fig.3.9. For instance, selecting $n = 0.75$ results in a 78.9% increase in TCP jerk with an associated 9.6% increase in peak synchronisation. Finally, it should be noted that the effect of peak synchronization on surface finish has not been related analytically in this work as there are other effects such tool geometry and cutting strategy, etc that influence the relationship. However, $|\lambda|_\infty$ still can be used as a heuristic metric to help tune the time constants.

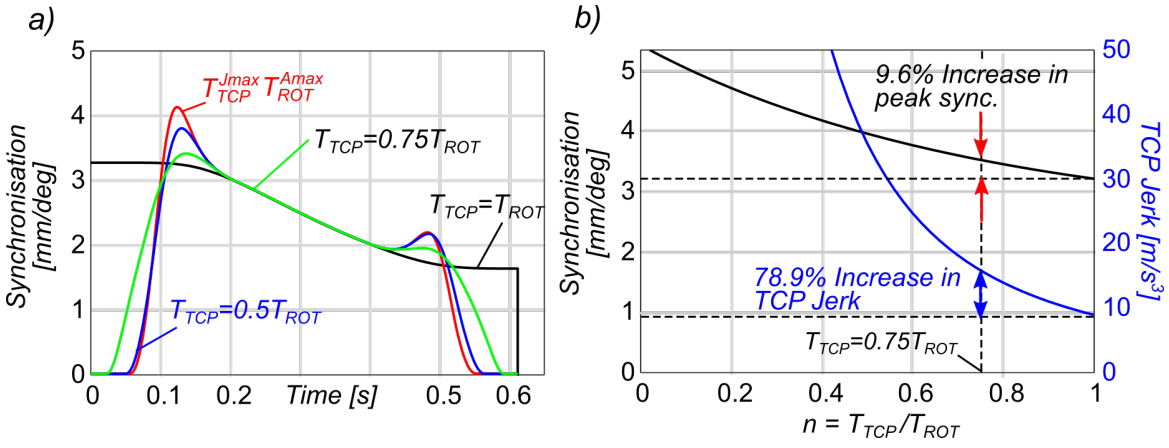

Figure 3.9: a) Synchronisation profiles b) Peak synchronisation and TCP Jerk

3.3 Non-stop Continuous Interpolation

As shown in the previous section, P2P motion can be interpolated accurately by the proposed joint MCS-WCS approach. In practice P2P interpolation is only used for a very limited number of finishing type 5-axis machining applications. In high-speed roughing and semi-finishing processes, non-stop continuous tool motion is required. This section extends the proposed interpolation approach to contouring toolpaths. A toolpath blending scheme with confined TCP and ORI interpolation errors is introduced. The non-stop interpolation approach presented in this section is inspired by the strategy presented in [48].

Consider a 5-axis toolpath commanded with 2 linear CL-lines (kth and $k+1$th) with identical lengths of L and at a feedrate of F (see Fig.3.10a). The TCP and rotary velocity pulses are commanded back-to-back, and they are filtered continuously as single velocity pulse stream. As shown, both the interpolated TCP and ORI trajectories deviate from the commanded CL-lines around the junction point, e.g. the corner, where the feed direction is altered. Interpolated TCP trajectory deviates from the commanded geometry in the WCS and the error is defined by the Euclidean distance. Whereas the angular blending error is measured in the spherical coordinates but controlled by the rotary axes of the 5-axis machine tool.

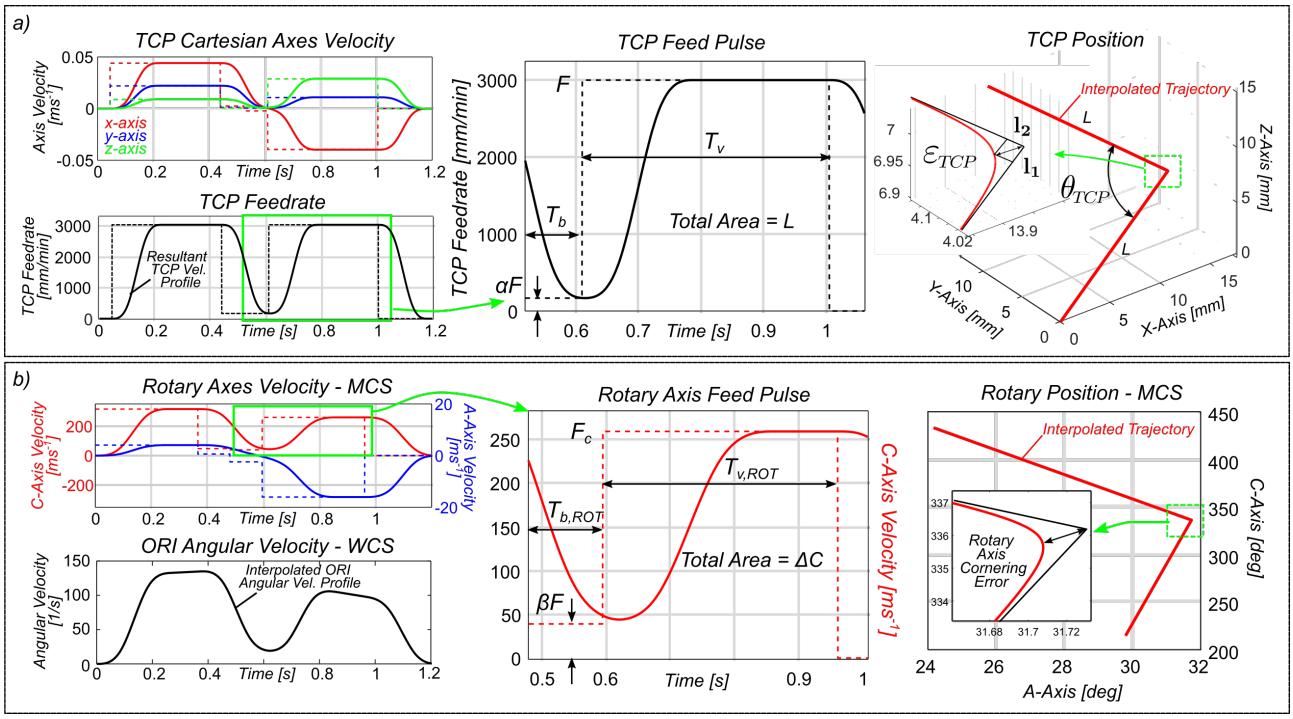

Figure 3.10: 6DOF corner motion commanded by a) Cartesian and b) rotary velocity blending pulses

The strategy is to lower the feedrate, around the corner so that the resultant TCP and ORI blending errors are confined by user set tolerances. The characterization and control of the TCP blending errors in the WCS are derived and reported in [48] and will only be briefly summarised here. In order to control the TCP blending errors, the tangential feedrate F is reduced locally around the corner by a feed override factor of α, for a period of:

$$T_b = \frac{1}{2}T_d(1-\alpha) = \frac{3}{2}T_{TCP}(1-\alpha), \tag{3.32}$$

where $T_d = 3T_{TCP}$ is the total filter delay. Notice that this approach is different than the one in [25] where cornering feed pulse duration is fixed to half of the FIR filter time constant. The minimum Euclidean distance and analytical TCP displacement equations form the following 5th order polynomial (derived in [48]):

$$FT_{TCP}\sqrt{\cos\left(\theta_{TCP}+1\right)}\left(-27\alpha^5+27\alpha^4+26\alpha\right)-128\sqrt{2}\varepsilon_{TCP}\leq 0. \tag{3.33}$$

Equation (3.33) can be solved for the real positive solution $\alpha \in [0,1]$ at any given cornering angle θ_{TCP}. Notice that the override factor α is also a function of the feedrate, F and the time constant T_{TCP}. Finally, when the feedrate of consecutive blocks are not identical the maximum cornering error does not occur in the middle of the cornering blend. However, if the feedrate F in (3.33) is selected to be the largest feed the actual blending error would be overestimated. The calculated feed override factor would ensure TCP tolerance would be met and thus provide a safe method of trajectory planning.

3.3.1 Control of ORI Blending Errors (blending errors in segment junction)

In the same manner as the TCP, non-stop interpolation of tool orientation also generates interpolation errors around segment junctions. Such tool orientation blending errors ε_{ORI} must be measured in the spherical coordinates. However, note that tool orientation is interpolated in the WCS by the rotary axes of the 5-axis machine tool. Figure 3.11 shows the rotary axes and tool trajectories in machine and workpiece coordinates. As shown in Fig.3.11b, due to low-pass filtering-based non-stop interpolation, rotary axis trajectories deviate from the junction point of the successive CL-lines in the MCS. As a result, the interpolated ORI vectors deviate from the commanded tool orientation in the spherical coordinates.

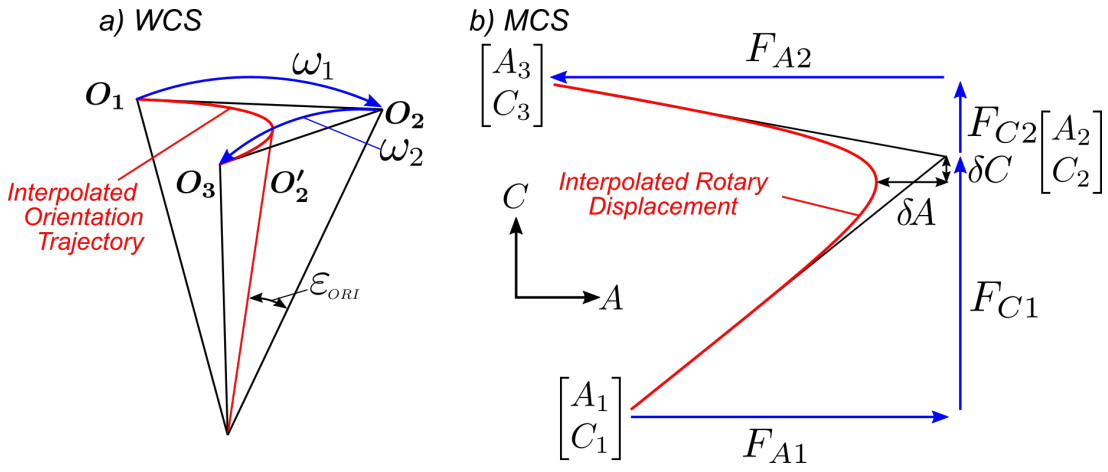

Figure 3.11: Equivalent WCS and MCS tool motion (red) and displacements (blue).

To control the ORI blending errors, the rotary axes feedrates (F_A, F_C) are reduced locally around the corner by the rotary feed override factor β as shown in Fig.3.10b. The duration of this lower "cornering" feed pulse is:

$$T_{b,ROT} = \frac{3}{2} T_{ROT}(1-\beta). \tag{3.34}$$

The rotary displacement $(\Delta A, \Delta C)$ for each axis is equal to the area underneath the velocity pulses (the C-axis displacement is shown as n highlighted example in Fig.3.10). Hence, the feed pulse duration $T_{v,ROT}$ is computed to satisfy the travel length condition by setting the pulse length as:

$$\Delta A = F_A T_{v,ROT} + \beta F_A T_{b,ROT} \rightarrow T_{v,ROT} = \frac{\Delta A}{F_A} - \beta T_{b,ROT}, \tag{3.35}$$

$$\Delta C = F_C T_{v,ROT} + \beta F_C T_{b,ROT} \rightarrow T_{v,ROT} = \frac{\Delta C}{F_C} - \beta T_{b,ROT}. \tag{3.36}$$

As shown in Fig.3.10, the rotary axis interpolation error can be calculated from the corner blending geometry based on the distances traveled during the deceleration and acceleration transients. These distances from the junction point can be computed from the components of the adjacent kinematic profiles as follows. First, the displacements for decelerating phases (s_{A1}, s_{C1}) and accelerating phases (s_{A2}, s_{C2}) for each axis are calculated from the interpolated rotary displacement profile by evaluating the piecewise analytical displacement equation (3.61) (see Appendix 3E) at $t = 3T_{ROT}/2$:

$$s_{\theta i} = \frac{F_{\theta i}}{128} T_{ROT} \left(-27\beta^5 + 27\beta^4 + 26\beta\right), \quad \theta = A, C, \quad i = 1, 2 \tag{3.37}$$

The total axis displacement error $(\delta A, \delta C)$ (see Fig.3.11) from the corner junction for each axis is a superposition of the decelerating and accelerating components such that:

$$\delta A = s_{A1} - s_{A2}, \tag{3.38}$$

$$\delta C = s_{C1} - s_{C2}. \tag{3.39}$$

Expressing (3.38) and (3.39) using (3.37) results in the following equations for rotary axis interpolation errors,

$$\delta A = \frac{T_{ROT}}{128} \left(F_{A2} - F_{A1}\right) \left(-27\beta^5 + 27\beta^4 + 26\beta\right), \tag{3.40}$$

$$\delta C = \frac{T_{ROT}}{128} \left(F_{C2} - F_{C1}\right) \left(-27\beta^5 + 27\beta^4 + 26\beta\right). \tag{3.41}$$

Next, the deviation of the rotary axis positions from the commanded CL-lines in the WCS is related to the angular deviation of the tool axis vector in the spherical coordinates as shown in Fig.3.11a.

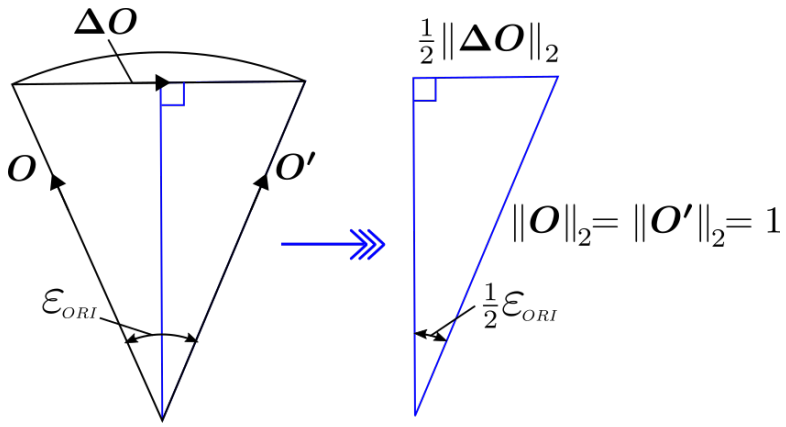

Figure 3.12: Tool Orientation Error in the WCS

Figure 3.12 illustrates the geometrical relationship between the interpolated ORI, $\mathbf{O}' = [O_i', O_j', O_k']^{\mathrm{T}}$, the commanded ORI $\mathbf{O} = [O_i', O_j', O_k']^{\mathrm{T}}$ and ε_{ORI}, which is the angular deviation, and ε_{ORI} can be related to the difference in the interpolated and commanded ORI vectors $\Delta\boldsymbol{O} = \boldsymbol{O} - \boldsymbol{O}'$ as:

$$\frac{1}{2}\varepsilon_{ORI} = \sin^{-1}\left(\frac{\frac{1}{2}\|\Delta\mathbf{O}\|_2}{\|\mathbf{O}'\|_2}\right), \tag{3.42}$$

$\boldsymbol{\Delta O}$ is then calculated from rotary axis errors in WCS by making use of the rotary Jacobian $\boldsymbol{J}_{ROT}$ through the following relationship:

$$\Delta\mathbf{O} \approx \mathbf{J}_{ROT}\Delta\boldsymbol{\theta}, \tag{3.43}$$

where $\boldsymbol{\Delta\theta} = [\delta A, \delta C]^T$ and is assumed small for the approximation to be valid. Hence, the tool orientation tolerance $\varepsilon_{ORI,Tol}$ is respected when

$$\|\mathbf{J}_{ROT}\Delta\boldsymbol{\theta}\|_2 \leq \sin\left(\varepsilon_{ORI,Tol}\right). \tag{3.44}$$

Eq.(3.44) directly relates the orientation error in WCS using rotary axis displacements in MCS. Substituting the expressions from Eqs.(3.40) and (3.41), and defining $\Delta F_A = F_{A2} - F_{A1}$ and $\Delta F_C = F_{C2} - F_{C1}$ (from Eq.(4.26)), leads to:

$$\left\|\boldsymbol{J}_{ROT}\begin{bmatrix}\Delta F_A\\ \Delta F_C\end{bmatrix}\right\|_2 \frac{T_{ROT}}{128}\left(-27\beta^5 + 27\beta^4 + 26\beta\right) - \sin\left(\varepsilon_{ORI,Tol}\right) \leq 0, \tag{3.45}$$

also notice that

$$\left\|\mathbf{J}_{ROT}\begin{bmatrix}\Delta F_A\\ \Delta F_C\end{bmatrix}\right\|_2 = \left\|\begin{bmatrix}\Delta\omega_i\\ \Delta\omega_j\\ \Delta\omega_k\end{bmatrix}\right\|_2 = \Delta\omega_{ORI}. \tag{3.46}$$

Eq.(3.45) represents the relationship between tool orientation tolerance $\varepsilon_{ORI,Tol}$, change in tool angular velocity $\Delta\omega_{ORI}$, rotary FIR filter time constant T_{ROT} and the reduction in rotary velocity during cornering transitions β. For a defined tool orientation tolerance and change in angular velocity, (3.45) can be solved for the real positive solution where $\beta = [0, 1]$. Effectively defining how much does the rotary axes need to slow down in the MCS to ensure the tool orientation tolerance in the WCS is satisfied. Fig.3.13 shows the rotary axis feed override factor β for 1 and 5 degree orientation tolerances with respect to change in angular velocity. These curves are used to select β when forming the rotary axes velocity pulse trains.

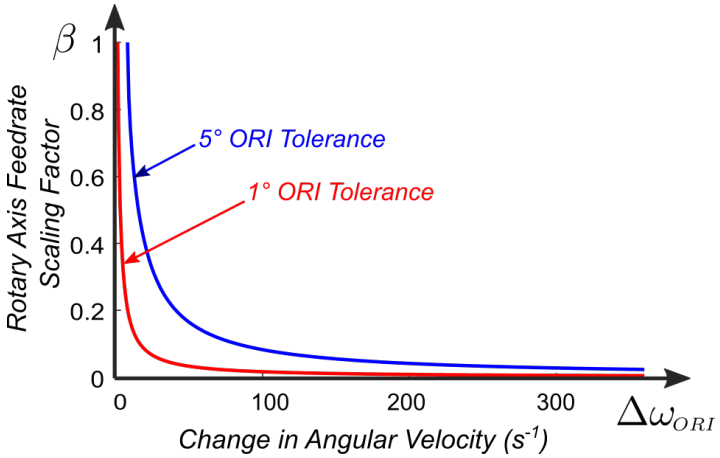

Figure 3.13: Reduction in rotary axis feedrate due to change in angular velocity for 1 and 5 degree tool orientation tolerances

3.4 Illustrative Examples

This section provides 2 simulation based case studies to demonstrate the effectiveness of the proposed FIR based interpolation method presented in section 3.3. The reference toolpath is commanded by the tool position and orientation commands given in Table 3.1. The commanded feedrate is 2000 mm/min and the TCP and ORI tolerances, ε_{TCP} ε_{ORI}, are set to 50μm and 1mrad respectively.

Px	Py	Pz	Oi	Oj	Ok
0	0	0	0.15	-0.1	0.9836
0	34.85	0	0.18	-0.08	0.9804
45	34.85	0	0.2	0.04	0.9790
75	0	0	0.14	0.02	0.9899
0	0	0	0.1	-0.05	0.9937

Table 3.1: Tool position and orientation commands for the illustrative example 5-axis toolpath

3.4.1 Control of Nonlinear Interpolation Error

In the first example, the reference toolpath is compared with the adaptively discritised toolpath using the method presented in section 3.2.2. The Cartesian FIR filter is set to match the rotary FIR filter, where $T_{TCP} = T_{ROT} = 0.102$s. Fig.3.14 shows the TCP and ORI trajectories for the two toolpaths, in particular 3.14c shows the deviation in the reference toolpath (blue line) from the absolute true path (black line) attributed to the nonlinear interpolation error. In Fig.3.14d the ORI contour error for the reference toolpath (shown in blue) exceeds the ε_{ORI}=1mrad tolerance. However, when using the proposed discritisation scheme, the modified toolpath satisfies the ORI tolerance throughout the tool motion. As only the ORI commands are adaptively modified, the effect on TCP accuracy is negligible as shown in Fig.3.14b.

3.4.2 Reducing Machining Cycle Times

Next, the effect of decoupling the cartesian TCP and rotary FIR filter time constants is demonstrated. Three cases are presented, which use the discritised toolpath from the previous example (See Fig.3.14). In the first case, Cartesian TCP and rotary FIR filter time constants are matched based on the maximum rotary axis acceleration constraint such that $T_{TCP} = T_{ROT} = 0.102$s. In the second case, the Cartesian FIR filter is set to 90% of the rotary FIR filter time constant,

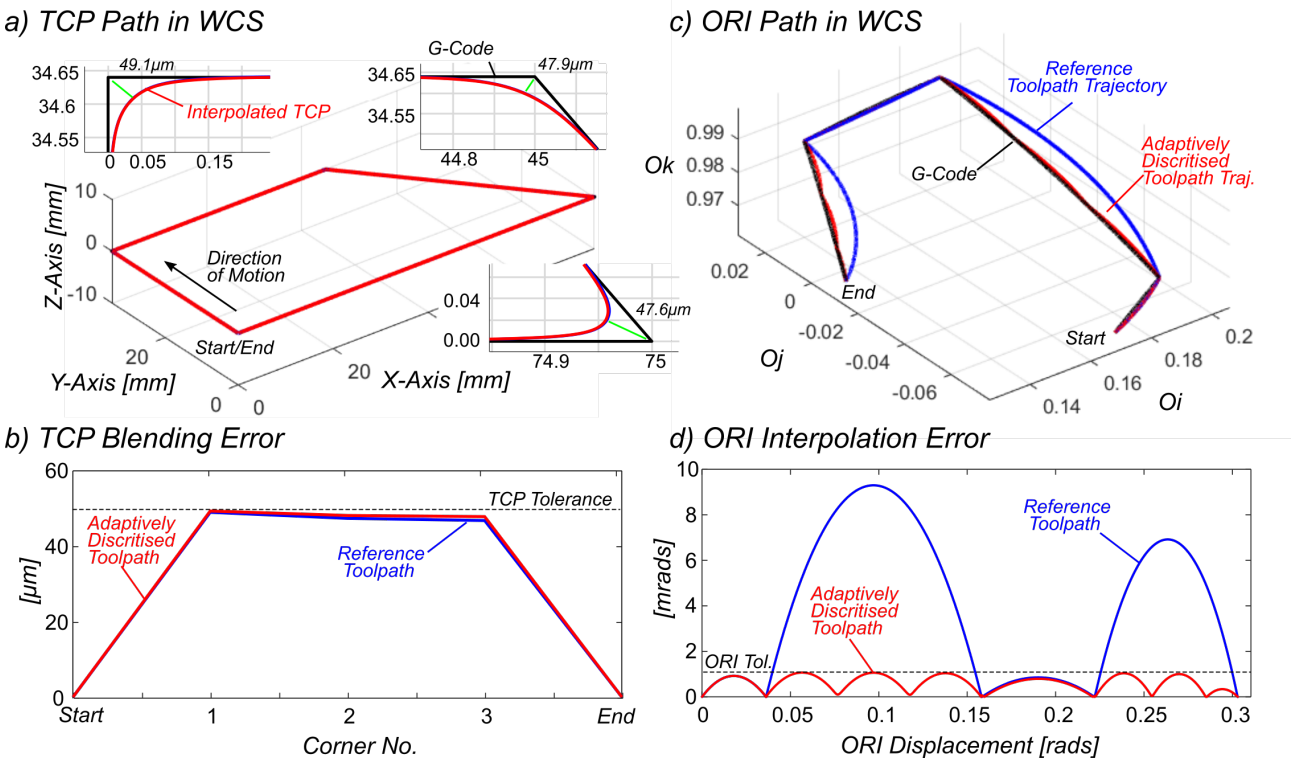

Figure 3.14: TCP and ORI trajectories and interpolation errors for reference and adaptively discritised toolpath.

$T_{TCP} = 0.9T_{ROT} = 0.092$s. Finally, in the last case, time constants are fully decoupled and individually set based on the maximum TCP jerk and maximum rotary acceleration, $T_{TCP} = 0.026$s and $T_{ROT} = 0.102$s. Note that Cartesian axes typically have significantly higher acceleration limits as compared to the rotary drives, and thus TCP jerk limits are hereby considered. Figure 3.15 shows our proposed method makes sure that TCP and ORI contouring tolerances are respected for all the cases. Figure 3.15b shows that even when the time constants are greatly decoupled, the TCP contouring error is kept below the user set contouring tolerance value. In terms of ORI contouring accuracy, Fig.3.15d shows the contouring errors are confined and well respected.

The interpolated kinematics are presented in Fig.3.16. The total machining cycle times are 7.21s,7.11s and 6.48s for cases 1,2 and 3 respectively. Case 2 shows a 1.4% reduction when compared to case 1, whilst case 3 shows an significant 10.1% reduction in cycle time. The significant difference is due to the exploitation of the TCP jerk limits, which can be seen in the jerk profiles for the 3 cases presented in Fig.3.16c. When the Cartesian FIR filter is set to match the rotary time constant (case 1), the maximum TCP jerk is less than $2.5\text{m}s^{-3}$. This results in slow TCP motion during the corner segments as seen in Fig.3.16b. By decoupling

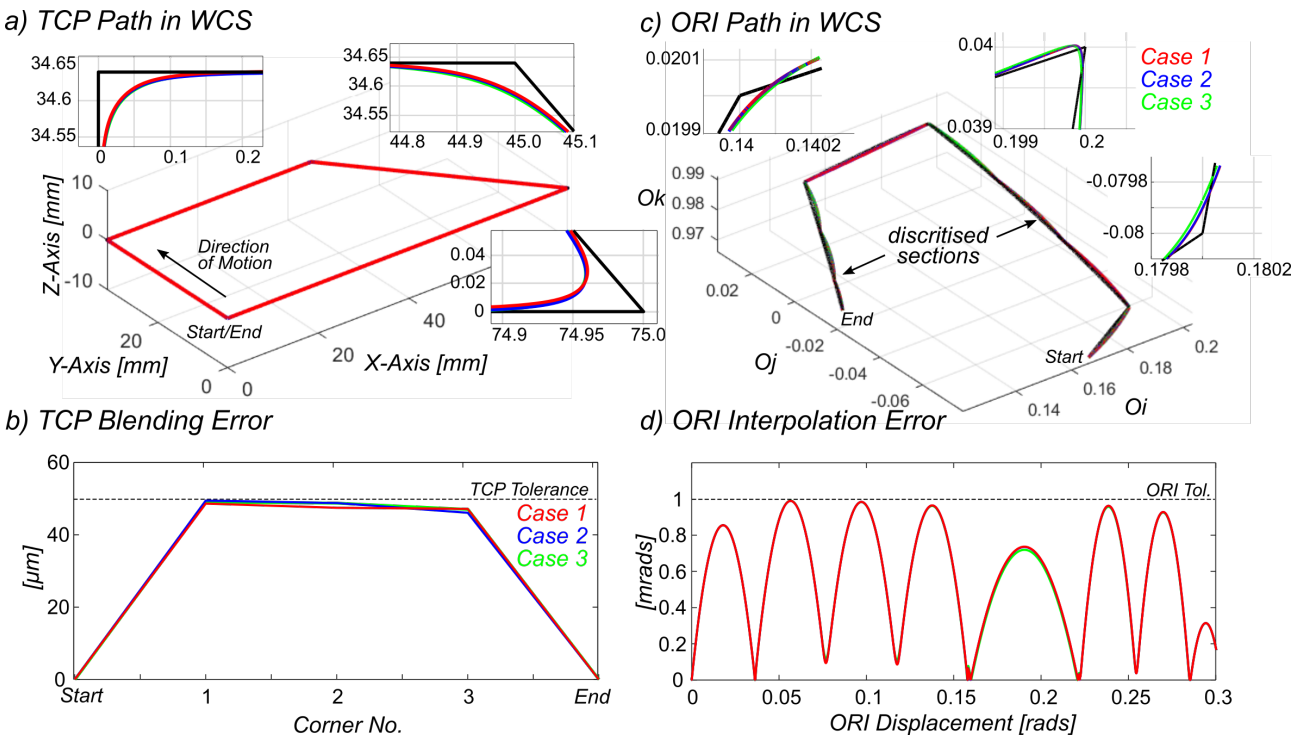

Figure 3.15: TCP and ORI trajectories and interpolation errors for the 3 illustrative cases

the Cartesian time constant and setting it based on maximum TCP jerk ($55 \mathrm{m} s^{-3}$), the maximum TCP acceleration is greater, the duration of cornering motion is shorter, and thus the overall machining time is reduced (as seen in the feedrate profile in Fig.3.16a). As the rotary FIR time constant is already set based on maximum rotary acceleration it cannot be lowered without violating their kinematic limits, which would likely induce larger ORI contouring errors.

Finally, synchronisation between ORI and TCP motion is shown in Fig.3.16g. The effect on synchronisation due to decoupled time constants is prominent at the start and end of tool motion. However, in practice, start and end of tool motion is never within the machining operation itself as entry/exit strategies such as rolling in/out are applied in 5-axis machining. Therefore, the proposed time constant decoupling approach presented in this example does not negatively affect the simultaneous motion synchronization of the 5-axis machine tool. It helps reduce cycle times significantly without deteriorating the contouring performance either.

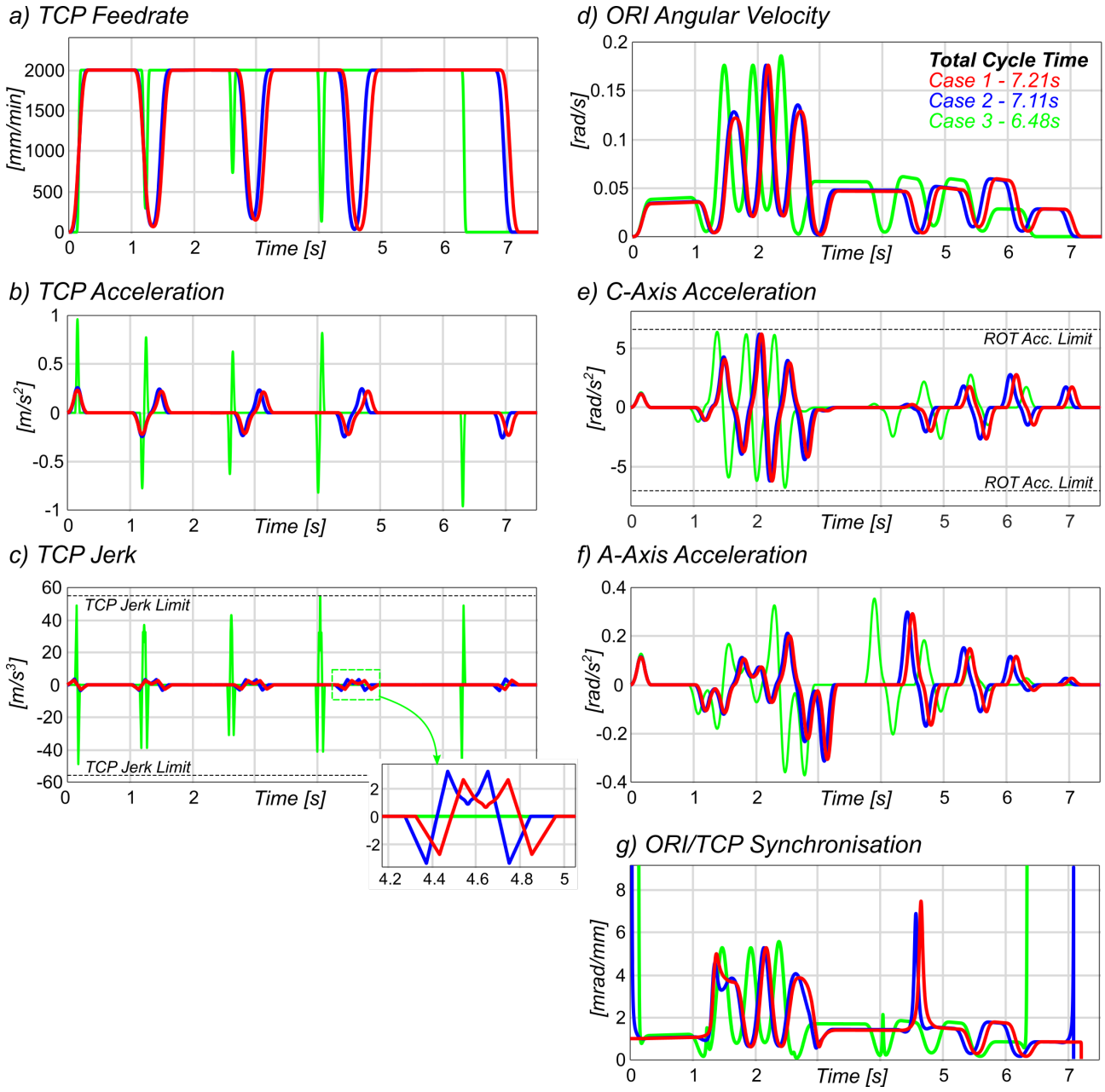

Figure 3.16: Interpolated TCP and ORI kinematic profiles along trapezium toolpath

3.5 Experimental Results

This section provides experimental validation case studies. The proposed interpolation method is experimentally verified on a 5-axis machining centre with rotary tilting table configuration. The machine is controlled by an open CNC system [45, 25] with a servo closed-loop sampling time set at 10kHz. The inverse kinematics transformation is given in Eq.(3.14). The reference toolpath is commanded by the tool position and orientation commands given in Table 3.2. The commanded feedrate is 2000 mm/min and the TCP and ORI tolerances are set to 50μm and 3mrad respectively.

Px	Py	Pz	Oi	Oj	Ok
20.8	0	0	0.611	0.272	0.7434
83.2	4.4	0	0.143	0.063	0.9877
78.6	24.8	0	0.127	0.092	0.9876
41.6	35.8	0	0.541	0.393	0.7436
79.8	59.0	0	0.448	0.497	0.7432
54.4	82.2	0	0.105	0.116	0.9877
18.2	81.4	0	0.064	0.143	0.9937

Table 3.2: Tool position and orientation commands for experimental 5-axis toolpath

The effect of decoupling the Cartesian TCP and rotary FIR filter time constants is demonstrated with two cases studies. In the first, Cartesian TCP and rotary FIR filter time constants are matched based on the maximum rotary axis acceleration constraint such that $T_{TCP} = T_{ROT} = 0.082$. In the second case, the time constants are fully decoupled and individually set based on the maximum TCP jerk and maximum rotary acceleration, $T_{TCP} = 0.026$ and $T_{ROT} = 0.082$. The measured TCP and ORI trajectories are presented in Figs.3.17a and 3.17c respectively. The TCP contouring errors in the local corners, shown in Fig.3.17b, respect the TCP tolerance. The ORI contouring errors are presented in Fig.3.17d where it is shown that both the cornering blending errors and nonlinear interpolation errors satisfy the ORI tolerance.

The servo following errors are presented for the X,Y,Z,A and C axis feed drives in Fig.3.18. There are no significant effects of decoupling the FIR filter time constants on the servo following error. In particular, the rotary drives show identical error profiles. The largest tracking error is observed in the Z-axis (see Fig.3.18c) at 3.8s which is linked to the TCP reaching jerk limit

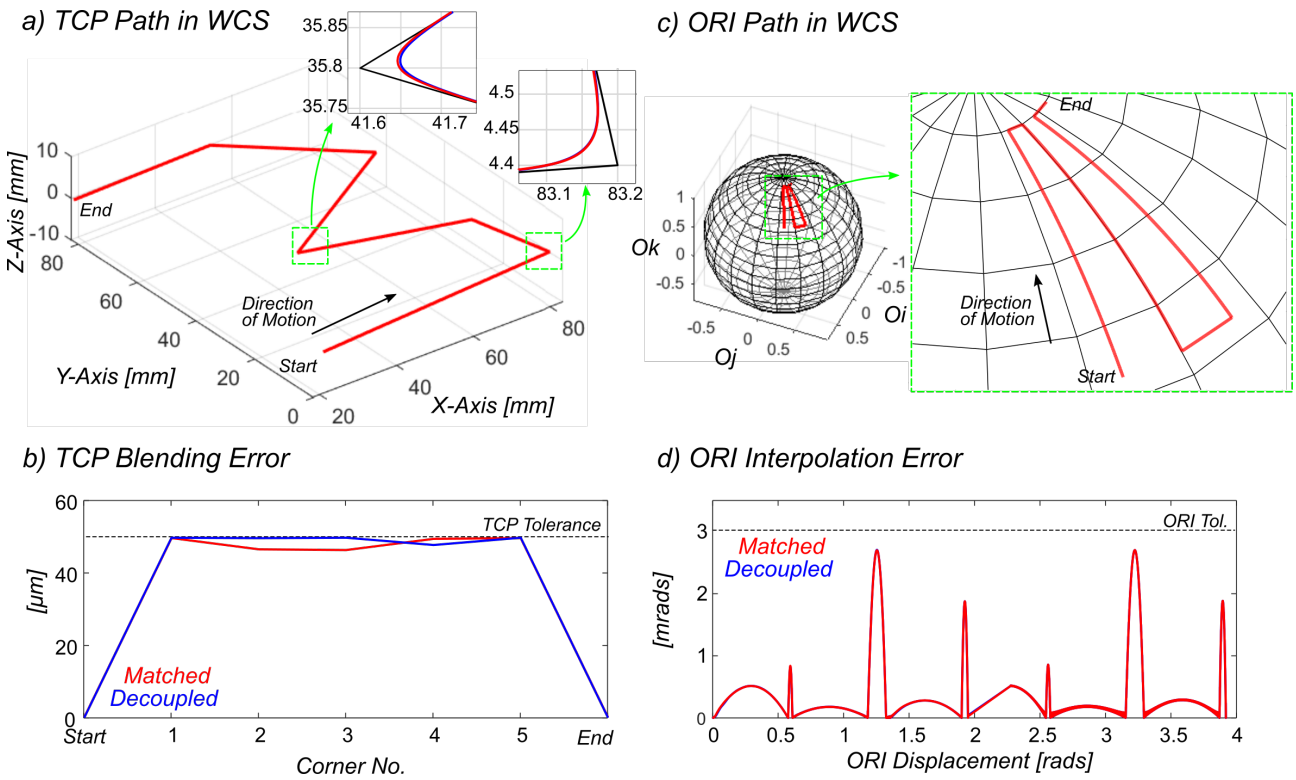

Figure 3.17: TCP and ORI trajectories and blending errors for experimental toolpath

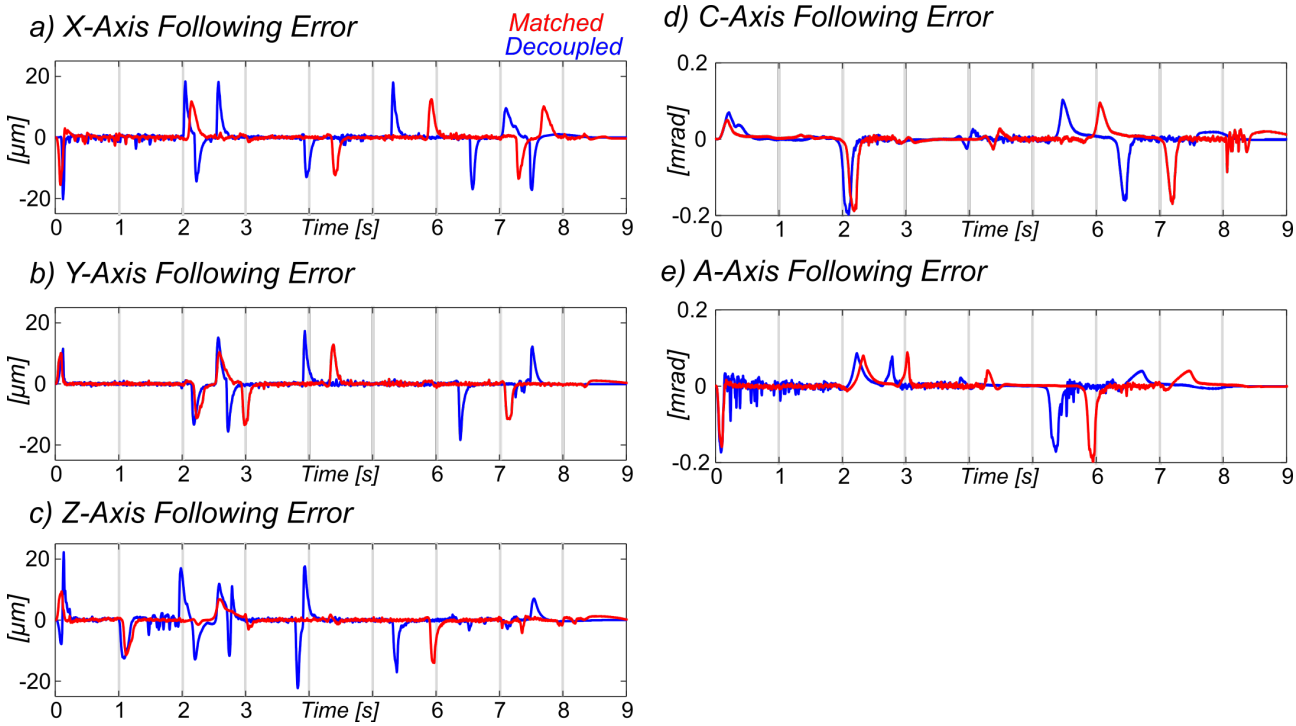

Figure 3.18: Servo following errors for X,Y,Z,A and C axis feed drives for matched (red) and decoupled (blue) FIR filter time constants

as shown in Fig.3.19c. This is also as expected, larger speeds and jerk values lead to larger tracking errors. Nevertheless, the difference is minor.

The measured kinematic profiles for the TCP and rotary axes are presented in Fig.3.19. The total machining cycle times are 8.41s and 7.58s for the matched and decoupled cases respectively. The decoupled case shows a 9.9% reduction in cycle time compared to the matched case. As with the illustrative example, the fundamental reason is the time saved during the cornering motions by exploiting the TCP jerk limit. In the matched case, the Cartesian FIR filter is matched to the rotary FIR filter, which is constrained by the rotary acceleration limit, 6.3 rads^{-2} (360°s^{-2}) as shown in Figs.3.19e and 3.19.f. Thus, the maximum TCP acceleration of the matched case is much lower than the decoupled case in Fig.3.19b. In the decoupled case, the Cartesian FIR filter time constant is set based on the maximum TCP jerk, 55 ms^{-1} as shown in Fig.3.19c. This results in faster acceleration and less time in the cornering motions and a reduced overall machining cycle time.

Finally, the synchronisation between TCP and ORI motion is shown in Fig.3.19g. For the decoupled case, it can be seen that there is no significant effect on tool synchronisation during the main part of the tool motion. To summarise, decoupling the FIR time constants maximises the performance of the machine by allowing the Cartesian axes to utilize their dynamic performance better, which significantly reduces the machining cycle time whilst satisfying TCP and ORI tolerances.

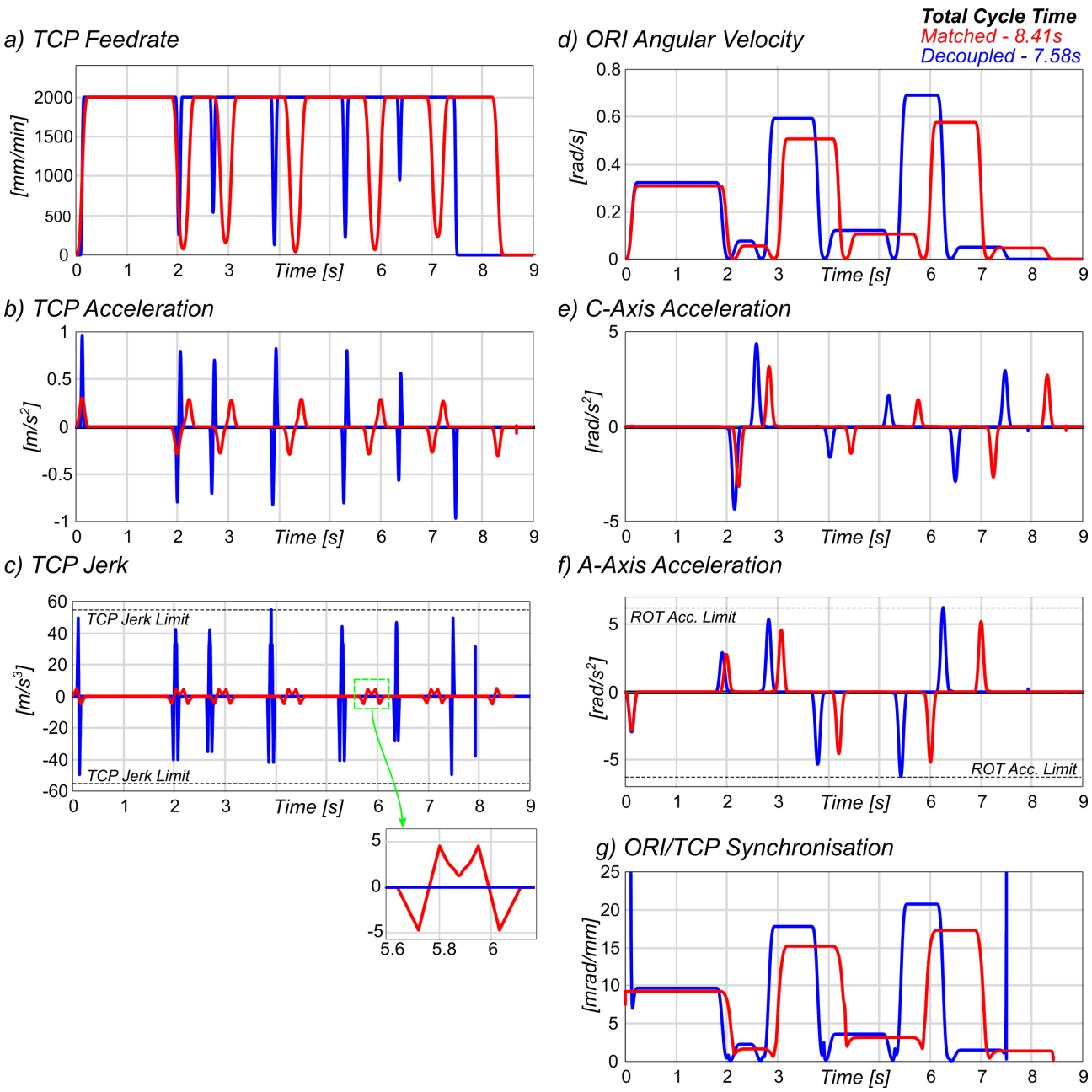

Figure 3.19: Interpolated TCP and ORI kinematic profiles along experimental toolpath

3.6 Conclusions

This paper presented a novel real-time interpolation technique for 5-axis machine tools. It introduced the use of a joint workpiece-machine coordinate system interpolation scheme based on finite impulse response filtering. The method eliminated the requirement for complex real-time spherical interpolation techniques while mitigating feed fluctuations when interpolating near kinematic singularities. A 10% reduction in machining cycle time was demonstrated by decoupling the Cartesian and rotary FIR filter time constants, maximising feed drive performance, whilst satisfying TCP and ORI user defined tolerances. The heuristic selection of the optimal FIR time constant was presented based on a characterised synchronisation between the TCP and ORI motions. To compensate for nonlinear tool orientation interpolation errors a new approach to adaptively discritising the ORI trajectory confining interpolation errors to within user defined tolerances was presented. Finally, blending errors during non-stop continuous interpolation of linear toolpaths were modelled and confined using the rotary feed override factor.

Appendix 3A Maximum nonlinear spherical geometry error occurs at u = 0.5

The start and end orientation vectors $[\mathbf{O_s}, \mathbf{O_e}]$ can be represented as a function of starting rotary positions $[A_s, C_s]$ and rotary displacements $[\Delta A, \Delta C]$:

$$\mathbf{O_s} = \begin{bmatrix} \sin A_s \sin C_s \\ -\sin A_s \cos C_s \\ \cos A_s \end{bmatrix}, \mathbf{O}_e = \begin{bmatrix} \sin\left(A_s + \Delta A\right) \sin\left(C_s + \Delta C\right) \\ -\sin\left(A_s + \Delta A\right) \cos\left(C_s + \Delta C\right) \\ \cos\left(A_s + \Delta A\right) \end{bmatrix}. \tag{3.47}$$

The spherically interpolated path in the WCS is:

$$\mathbf{O_{WCS}}\left(u\right) = \frac{\mathbf{O_s} \sin\left(\left(1-u\right)\theta\right) + \mathbf{O_e} \sin u\theta}{\sin\theta}, \tag{3.48}$$

and the MCS interpolated trajectory is defined by:

$$\mathbf{O_{MCS}}\left(u\right) = \begin{bmatrix} \sin\left(A_s + u\Delta A\right) \sin\left(C_s + u\Delta C\right) \\ -\sin\left(A_s + u\Delta A\right) \cos\left(C_s + u\Delta C\right) \\ \cos\left(A_s + u\Delta A\right) \end{bmatrix}. \tag{3.49}$$

Using the dot product to calculate the nonlinear interpolation error between the two interpolated trajectories results in:

$$\begin{aligned} \varepsilon_{MCS}(u) &= \cos^{-1}\left(\frac{\mathbf{O}_{MCS}(u) \cdot \mathbf{O}_{WCS}(u)}{\|\mathbf{O}_{MCS}(u)\| \, \mathbf{O}_{WCS}(u)\|}\right) \\ &= \cos^{-1}\left(\frac{\sin\left(A_s + u\Delta\mathrm{A}\right) \sin\left(C_s + u\Delta\mathrm{C}\right) \left(\sin\left(A_s + \Delta\mathrm{A}\right) \sin\left(C_s + \Delta\mathrm{C}\right) \sin(u\theta) - \sin(\theta(u-1)) \sin A_s \sin C_s\right)}{\sin\theta} + \ldots\right. \\ &\ldots - \frac{\cos\left(A_s + u\Delta\mathrm{A}\right) \left(\sin(\theta(u-1)) \cos A_s - \cos\left(A_s + \Delta\mathrm{A}\right) \sin(\theta u)\right)}{\sin(\theta)} + \ldots \\ &\ldots + \left.\frac{\cos\left(C_s + u\Delta\mathrm{C}\right) \sin\left(A_s + u\Delta\mathrm{A}\right) \left(\cos\left(C_s + \Delta\mathrm{C}\right) \sin\left(A_s + \Delta\mathrm{A}\right) \sin(u\theta) - \sin(\theta(u-1)) \cos C_s \sin A_s\right)}{\sin\theta}\right) = 0, \end{aligned} \tag{3.50}$$

$$\frac{d\varepsilon_{\mathrm{MCS}}}{du} = 0. \tag{3.51}$$

The derivative of the nonlinear interpolation error (3.50) is solved numerically. As shown in Fig.3.20 by setting $d\varepsilon_{MCS}/du = 0$, the maximum error occurs right in the middle of the trajectory for $u = 0.5$.

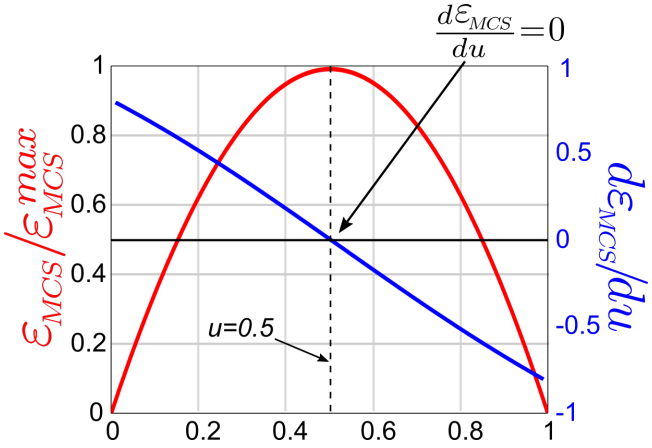

Figure 3.20: Nonlinear interpolation error and derivative showing maximum error occurs at u=0.5

Appendix 3B Haversine Equations for Arc Lengths

$$h_{BC} = \frac{1-\cos(-\Delta A)}{2} + \sin(A_s)\left(\sin(A_s)\cos(-\Delta A) - \cos(A_s)\sin(-\Delta A)\right)\frac{1-\cos(\Delta C)}{2}, \quad (3.52)$$

$$h_{AC} = \frac{1-\cos\left(\frac{-\Delta A}{2}\right)}{2} + \sin(A_s)\left(\sin(A_s)\cos\left(\frac{-\Delta A}{2}\right) - \cos(A_s)\sin\left(\frac{-\Delta A}{2}\right)\right)\frac{1-\cos\left(\frac{\Delta C}{2}\right)}{2}, \quad (3.53)$$

and

$$h_{AB} = \frac{1-\cos\left(\frac{-\Delta A}{2}\right)}{2} + \sin\left(A_s+\frac{\Delta A}{2}\right)\left(\sin\left(A_s+\frac{\Delta A}{2}\right)\cos\left(\frac{-\Delta A}{2}\right) - \cos(A_s)\sin\left(\frac{-\Delta A}{2}\right)\right)\frac{1-\cos\left(\frac{\Delta C}{2}\right)}{2}. \quad (3.54)$$

Appendix 3C Rotary and TCP's Cartesian Kinematic Profiles for P2P Motion with Alignment

$$v_\theta(t) = \begin{cases} \frac{F_\theta t^3}{6{T_{ROT}}^3}, & 0 \le t < T_{ROT} \\ \frac{F_\theta}{2{T_{ROT}}^3}\left({T_{ROT}}^3 - 3{T_{ROT}}^2 t + 3T_{ROT}t^2 - \frac{2t^3}{3}\right), & T_{ROT} \le t < 2T_{ROT} \\ \frac{F_\theta}{2{T_{ROT}}^3}\left(-7{T_{ROT}}^3 + 9{T_{ROT}}^2 t - 3T_{ROT}t^2 + \frac{t^3}{3}\right), & 2T_{ROT} \le t < 3T_{ROT} \\ F_\theta, & 3T_{ROT} \le t < T_v \end{cases} \tag{3.55}$$

$$v_{TCP}(t) = \begin{cases} \frac{F}{48{T_{TCP}}^3}(3T_{TCP} - 3T_{ROT} + 2t)^3, & 0 \le t < T_{TCP} \\ \frac{F}{2} + \frac{F}{24{T_{TCP}}^3}\left((3T_{ROT} - 2t)^2 - 9{T_{TCP}}^2\right)(3T_{ROT} - 2t), & T_{TCP} \le t < 2T_{TCP} \\ F - \frac{F}{48{T_{TCP}}^3}(3T_{TCP} + 3T_{ROT} - 2t)^3, & 2T_{TCP} \le t < 3T_{TCP} \\ F, & 3T_{TCP} \le t < T_v \end{cases} \tag{3.56}$$

$$s_\theta(t) = \begin{cases} \frac{F_\theta\, t^4}{24\, T_{ROT}^3}, & 0 \le t < T_{ROT} \\ -\frac{F_\theta}{24\, T_{ROT}^3}\,\left(3\, T_{ROT}^4 - 12\, T_{ROT}^3\, t + 18\, T_{ROT}^2\, t^2 - 12\, T_{ROT}\, t^3 + 2\, t^4\right), & T_{ROT} \le t < 2T_{ROT} \\ \frac{F_\theta}{24\, T_{ROT}^3}\,\left(45\, T_{ROT}^4 - 84\, T_{ROT}^3\, t + 54\, T_{ROT}^2\, t^2 - 12\, T_{ROT}\, t^3 + t^4\right), & 2T_{ROT} \le t < 3T_{ROT} \\ F_\theta\, t - \frac{3\, F_\theta\, T_{ROT}}{2}, & 3T_{ROT} \le t < T_v \end{cases} \tag{3.57}$$

Appendix 3D Derivation of Peak Synchronisation

The synchronisation peak λ_∞ is calculated by solving (3.30). Where v'_A,v'_C,v'_x,v'_y and v'_z are solved by evaluating the aligned analytical P2P Cartesian and rotary velocity equations (3.55) and (3.56) at $t = \frac{1}{2}(3T_{ROT} + T_{TCP})$, resulting in:

$$v'_i|_{t=\frac{1}{2}(3T_{ROT}+T_{TCP})} = \frac{F_i}{24T_{ROT}^3}\left(-{T_{TCP}}^3 + 9\,T_{TCP}\,T_{ROT}^2 + 12\,T_{ROT}^3\right), \quad i = [A, C] \tag{3.58}$$

$$v'_j\big|_{t=\frac{1}{2}(3T_{ROT}+T_{TCP})} = \frac{5F_j}{6}, \quad j = [x, y, z] \tag{3.59}$$

similarly, A results from solving the aligned displacement equation (3.57) at $t = \frac{1}{2}(3T_{ROT} + T_{TCP})$:

$$A'|_{t=\frac{1}{2}(3T_{ROT}+T_{TCP})} = \frac{F_A}{192\,T_{ROT}^3}\left(-{T_{TCP}}^4 + 18\,{T_{TCP}}^2\,T_{ROT}^2 + 48\,T_{TCP}\,T_{ROT}^3 + 39\,T_{ROT}^4\right). \tag{3.60}$$

Appendix 3E Axis Blending Pulse Kinematic Profiles

First three kinematic sections only presented for brevity, where $i = XYZAC$ and $\gamma = \alpha$, $T = T_{TCP}$ and $T_b = T_b$ for i = XYZ, and $\gamma = \beta$, $T = T_{ROT}$ and $T_b = T_{b,ROT}$for $i = AC$.

$$s'(t) = \begin{cases} \frac{F_i \gamma t^4}{24T^4}, & 0 \le t < T \\ -\frac{F\gamma}{24\,T^3}\,\left(3\,T^4 - 12\,T^3\,t + 18\,T^2\,t^2 - 12\,T\,t^3 + 2\,t^4\right), & T \le t < T_b \\ -\frac{F}{384}\left(81\,T^4\,\gamma^5 - 405\,T^4\,\gamma^4 + 810\,T^4\,\gamma^3 - 810\,T^4\,\gamma^2 + \ldots\right. & \\ \ldots + 453\,T^4\,\gamma - 81\,T^4 + 216\,T^3\,\gamma^4\,t - 864\,T^3\,\gamma^3\,t + \ldots & \\ \ldots + 1296\,T^3\,\gamma^2\,t - 1056\,T^3\,\gamma\,t + 216\,T^3\,t + 216\,T^2\,\gamma^3\,t^2 + \ldots & \\ \ldots - 648\,T^2\,\gamma^2\,t^2 + 936\,T^2\,\gamma\,t^2 - 216\,T^2\,t^2 + 96\,T\,\gamma^2\,t^3 + \ldots & \\ \left.\ldots - 384\,T\,\gamma\,t^3 + 96\,T\,t^3 + 48\,\gamma\,t^4 - 16\,t^4\right), & T_b \le t < 2T \end{cases} \tag{3.61}$$

$$v'(t) = \begin{cases} \frac{F_i \gamma t^3}{6T^3}, & 0 \le t < T \\ \frac{F_i\,\gamma}{2} + \frac{\gamma F_i}{24\,T^3}\,\left(\left(3\,T - 2\,t\right)^2 - 9\,T^2\right)\,\left(3\,T - 2\,t\right), & T \le t < T_b \\ -\frac{F_i}{48T^3}(27\,T^3\,\gamma^4 - 108\,T^3\,\gamma^3 + 162\,T^3\,\gamma^2 + \ldots & \\ \ldots - \ 32\,T^3\,\gamma + 27\,T^3 + 54\,T^2\,\gamma^3\,t - 162\,T^2\,\gamma^2\,t + \ldots & \\ \ldots + 234\,T^2\,\gamma\,t - 54\,T^2\,t + 36\,T\,\gamma^2\,t^2 - 144\,T\,\gamma\,t^2 + \ldots & \\ \ldots + 36\,T\,t^2 + 24\,\gamma\,t^3 - 8\,t^3), & T_b \le t < 2T \end{cases} \tag{3.62}$$

$$a'(t) = \begin{cases} \frac{F_i \gamma t^2}{2T^3}, & 0 \le t < T \\ -\frac{F\gamma}{2\,T^3}\,\left(3\,T^2 - 6\,T\,t + 2\,t^2\right), & T \le t < T_b \\ -\frac{F_i}{8T^3}(9\,T^2\,\gamma^3 - 27\,T^2\,\gamma^2 + 39\,T^2\,\gamma - 9\,T^2 + 12\,T\,\gamma^2\,t + \ldots & \\ \ldots - 48\,T\,\gamma\,t + 12\,T\,t + 12\,\gamma\,t^2 - 4\,t^2), & T_b \le t < 2T \end{cases} \tag{3.63}$$

$$j'(t) = \begin{cases} \frac{\gamma F_i t}{T^3}, & 0 \le t < T \\ \frac{F\gamma}{T^3}\,\left(3\,T - 2\,t\right), & T \le t < T_b \\ -\frac{\gamma F_i}{2T^3}\,\left(3\,T - 2\,t - 12\,T\,\gamma + 6\,\gamma\,t + 3\,T\,\gamma^2\right), & T_b \le t < 2T \end{cases} \tag{3.64}$$

Chapter 4

Journal Paper 3 - Optimising Machining Cycles Times using On-The-Fly Trajectory Generation

4.1 Introduction

The function and capability of the interpolator in computer numerical control (CNC) machining is vital to the performance of the overall machining process. It is the interpolator which determines the overall machining cycle time by generating synchronised feed drive reference commands. Traditional polynomial based methods of NC interpolation [10, 13], which are limited in their real-time processing capabilities, are being replaced in commercial machining centres with computationally efficient filter based algorithms [15, 24]. This paper further pushes the boundaries of FIR-based NC interpolation and in particular reduces overall machining cycle times through on-thy-fly interpolation.

The application of FIR filtering as an efficient means of NC interpolation was first introduced by Tajima et al. [21]. The research showed a chain of FIR filters can effectively smooth velocity pulse commands, which once integrated, generate smooth axis position commands. The TCP blending error was satisfied by controlling the overlap time between adjacent interpolated velocity profiles. The authors extended the method to 5-axis machining [25] and demonstrated tool centre point (TCP) and tool orientation (ORI) blending error control during corner segments analytically through dwell and velocity control. Liu et al. applied FIR filtering to 5-axis NC interpolation and incorporated both Cartesian and rotary axis limits which were satisfied

using a prediction and correction scheme [49]. Ward et al. further extended the use of velocity blending pulses with FIR based interpolation [48]. The method analytically constrained and controlled control cornering feedrates and TCP blending tolerance. The research also used a chain of FIR filters with matching time constants to simplify the selection of the time constant to a single parameter based on drive kinematic constraints and maximum feedrate. Extending to 5-axis, the velocity blending pulse approach was applied to a joint workpiece coordinate system-machine coordinate system, (WCS-MCS) interpolation scheme which satisfied tool orientation blending tolerance by controlling the rotary axis cornering feedrates [61]. Jiang et al. adopted a two step approach to 5-axis NC interpolation [50]. First, by inserting two B-splines to the tool tip trajectory and the tool orientation trajectory which each controls the smoothing errors independently followed by FIR filtering considering both tangential and axial kinematic constraints. Further work using spline based interpolation was presented by Song et al. [62]. Using position curvature information, a discontinuous feedrate profile was generated. Using two cascaded FIR filters, the axial velocities were smoothed and bounded to meet tangential jerk limits. A FIR filtering contour error (FICE) was introduced by their method and therefore a compensation method was integrated within the feedrate scheduling to overcome. Finally, focusing on global corners which are composed of highly discritised short CL-lines Tajima et al. [51] used a windowing based FIR-filtering method to control TCP and ORI blending tolerances.

The common approach to all FIR-based NC interpolation research to date is the FIR filter time constant does not change throughout the toolpath. The methods use either two or three cascaded FIR filters depending on the required order of the interpolated output signal, and the time constants are selected to meet the kinematic constraints of the machine tool feed drives. However, all assume a constant or worse case maximum commanded feedrate to design the time constants. This does indeed ensure the peak acceleration and jerk do not breach the kinematic limits throughout the toolpath (during P2P machining). However, this comes at a cost of the kinematic performance being sub-optimal for all motions except those commanded at the maximum feedrate. The result is longer overall machining cycles times than is necessary. This research overcomes this major limitation of standard FIR based NC interpolation.

For the first time, a novel on-the-fly (OTF) based FIR interpolation method is presented. The OTF NC interpolation method individually optimises the kinematic profiles of each individual G01 command online by filtering each CL-line separately based on the local feedrate command. The TCP and tool orientation (ORI) blending errors are controlled through a signal recon-

struction method termed Overlap and Add (OLA). Reductions in overall machining cycle time are gained for both point-to-point (P2P) and non-stop continuous machining.

The chapter is structured as follows. First, OTF interpolation is introduced for 3-axis P2P motion. Second, the method is extended to 3-axis continuous motion whilst addressing methods of jerk control. This is followed by OTF interpolation of 5-axis P2P and continuous motion with a new method of relating tool orientation blending errors in the MCS to the WCS when using dwell to control tolerance. Finally, simulations are conducted to demonstrate the efficacy of the OTF methods.

4.2 3-Axis On-The-Fly Interpolation

The key concept of the presented OTF interpolation method is each individual G01 command is filtered and integrated separately to generate segmented kinematically optimised smooth position commands which are reconstructed to form the full feed drive position reference commands. The following sections demonstrate the method to interpolate the segmented G01 commands/CL-lines.

4.2.1 3-Axis TCP Interpolation

Using subscript i to denote the i^{th} G01 command where $i = [1, ..., N_{G01}]$ and N_{G01} is the number of G01 commands, we define the P2P linear (G01) commands as the tool motion from the starting TCP positions $\mathbf{P}_i = [P_{i,x}, P_{i,y}, P_{i,z}]^T$ to the final tool position defined by $\mathbf{P}_{i+1}$. The machining feedrate F_i is defined as the linear speed of the TCP which can be presented by a single feed pulse $v_{TCP,i}(t)$:

$$v_{TCP,i}(t) = \begin{cases} F_i, & 0 < t \leq T_{v,i} \\ 0, & t > T_{v,i} \end{cases} \tag{4.1}$$

where $L_i = \|\mathbf{P}_{i+1} - \mathbf{P}_i\|_2$ and the duration of the feed pulse is calculated as $T_{v,i} = L_i/F_i$. Next, the tangential feed pulse $v_i(t)$ is resolved into its Cartesian TCP velocity pulse components:

$$\begin{bmatrix} v_{x,i}(t) \\ v_{y,i}(t) \\ v_{z,i}(t) \end{bmatrix} = v_{TCP,i}(t) \cdot \mathbf{u}_i, \text{ where } \mathbf{u}_i = \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{L_i}, \tag{4.2}$$

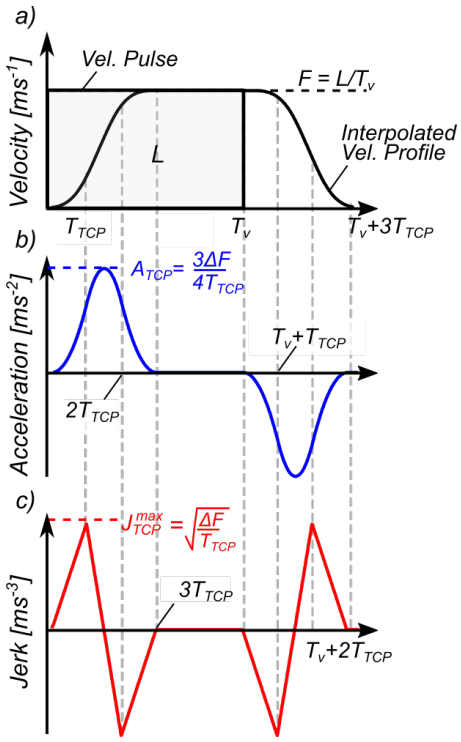

Figure 4.1: Kinematic profiles for a velocity pulse interpolated with 3 matching FIR filter

and the TCP is interpolated linearly in the WCS by filtering the Cartesian velocity pulses by a low-pass FIR filter as:

$$\begin{bmatrix} v'_{x,i}(t) \\ v'_{y,i}(t) \\ v'_{z,i}(t) \end{bmatrix} = v(t)_{TCP,i} \cdot \mathbf{u}_i * m_{TCP,i}(t), \tag{4.3}$$

where * is the convolution operation, $m_{TCP,i}(t)$ is the impulse response of the FIR filter and $[v'_{x,i}, v'_{y,i}, v'_{z,i}]$ are the filtered velocity signals.

Setting the time constants of 3 chained FIR filters identical [48] (i.e. matching) the resulting interpolated velocity signal represents a bell-shaped acceleration profile and a triangular jerk profile (see Fig. 4.1). The piecewise analytical equations for the kinematic profiles are obtained by solving the convolution integral in Eqn.(4.3), which are presented in appendix A.
The TCP peak tangential acceleration and jerk values are determined from the commanded feedrate F and FIR filter time constant T_{TCP}. As shown in 4.1b 4.1c respectively, when using 3 FIR filters with matching time constants, the maximum acceleration and jerk peaks occur at $t = 3T_{TCP}/2$ and at $t = T_{TCP}$ respectively. As presented in chapter 3, T_{TCP} can be selected to satisfy both the acceleration and jerk limits of the TCP motion:

$$T_{TCP} = \max\left\{\frac{3\Delta F}{4A_{TCP}^{\max}}, \sqrt{\frac{\Delta F}{J_{TCP}^{\max}}}\right\}. \tag{4.4}$$

Eq.(4.4) can be used to tune the filter time constants considering the largest change in feedrate ΔF throughout the entire tool-path. This would be the worst-case scenario-based tuning, leading to under-utilization of machine potential and elongated cycle times. In the proposed method, as each commanded CL-line is filtered separately, the FIR filter time constants are selected based on the local maximum axis feedrate command F_i:

$$T_{TCP,i} = \max\left\{\frac{3\Delta F_i}{4A_{TCP}^{\max}}, \sqrt{\frac{\Delta F_i}{J_{TCP}^{\max}}}\right\}. \tag{4.5}$$

where $\Delta F_i = \max\left(v_{TCP,i}(t) \cdot \mathbf{u}_i\right)$ presented in Eq.(4.2). The additional benefit of the proposed OTF method is that the acceleration or jerk limit can be modified online, providing capability for an adaptive acceleration or jerk limit.

Following FIR interpolation, the filtered TCP velocity signals are integrated to generate the segmented TCP position commands:

$$\begin{bmatrix} x_i'(t) \\ y_i'(t) \\ z_i'(t) \end{bmatrix} = \begin{bmatrix} P_{x,i} \\ P_{y,i} \\ P_{z,i} \end{bmatrix} + \int_0^t \begin{bmatrix} v_{x,i}'(t) \\ v_{y,i}'(t) \\ v_{z,i}'(t) \end{bmatrix} d\tau. \tag{4.6}$$

This section presented a method to segment the long position command signal into smaller segments enabling optimised FIR filtering for each separate G01 command. The next section presents the OLA method applied to generate P2P machining reference commands.

4.2.2 OTF Trajectory Generation of 3-Axis P2P Motion

The segmented smoothed position commands (4.6) can now be joined together to generate the full filtered feed drive reference command signals. Integrating the segmented velocity signals (4.2) to position signals prior to signal reconstruction increases the TCP accuracy over the full toolpath by eliminating errors associated with integrating long velocity signals in the time domain. For P2P toolpaths, the OLA method of signal reconstruction can be used to simply concatenate the segmented smoothed position command signals (i.e. overlap = zero) which would ensure an instantaneous stop in between each commanded CL-line. The interpolated TCP position commands for the full signal can be represented by:

$$\begin{bmatrix} x'(t) \\ y'(t) \\ z'(t) \end{bmatrix} = \begin{bmatrix} x'_i(t), x'_{i+1}(t), ..., x'_{N_{G01}}(t) \\ y'_i(t), y'_{i+1}(t), ..., y'_{N_{G01}}(t) \\ z'_i(t), z'_{i+1}(t), ..., z'_{N_{G01}}(t) \end{bmatrix}. \tag{4.7}$$

Next, an illustrative example is presented to show the benefits of OTF interpolation for P2P toolpaths.

Illustrative Example

Figure 4.2 shows a lock shaped toolpath [14] and the associated G-code commands. In this example, the feedrate is set at 3000 mm/min around the square section and lowers to 1500 mm/min around the arc. Three cases are presented, the first uses standard P2P FIR interpolation and the second and third both use the proposed P2P OTF interpolation method. In cases one and two the jerk limit is set at $50ms^{-3}$ and in the final case, the OTF method demonstrates an adaptive jerk limit which lowers to $20ms^{-3}$ around the arc.

Figure 4.3a shows the kinematic profiles for the standard FIR interpolation case. The FIR filter time constant is constant throughout the toolpath and is chosen based on the maximum feedrate and jerk (based on Eq.(4.4), $T_{TCP} = 0.0316$). It can be seen that the jerk limit is not exceeded during the P2P motion, but the toolpath is only jerk optimised for the faster motions, from approximately 2.5-4.5 seconds the axis jerk is around 50% of the limit. Investigating the OTF interpolation jerk profile (see Fig.4.3b) it can be seen that the full range of the jerk limit has been used throughout the toolpath. This is due to the OTF time constant based on Eq.(4.5) optimising each segment based on the maximum axis feedrate. The axis acceleration profiles are visibly higher for the OTF method compared to the standard FIR method. The result is

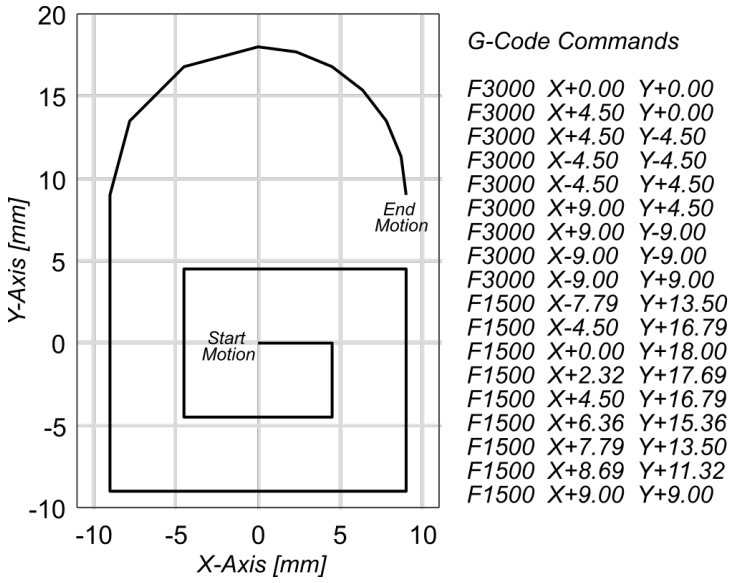

Figure 4.2: Lock toolpath and G-code commands

a reduction in overall machining cycle time. The total machining cycle time for case one is 4.56s and compared to the faster OTF method (case two) of 4.27s is a 6.8% reduction in time. The machining cycle time can be further reduced during continuous machining which will be demonstrated later in this paper.

Finally, case 3 demonstrates the adaptive nature of the OTF method and shows the acceleration and jerk limit can be changed online. In Fig.4.3c, the jerk limit is lowered around the arc of the toolpath. The jerk profile shows the axis jerk satisfying both the higher and lower user defined limits at different stages along the toolpath. This section has demonstrated the benefits of using OTF interpolation for P2P machining, however, most machining operations use continuous non-stop motion. The following section will show how the overlap between successive segmented signals can control TCP interpolation errors and peak axis jerk during continuous machining toolpaths.

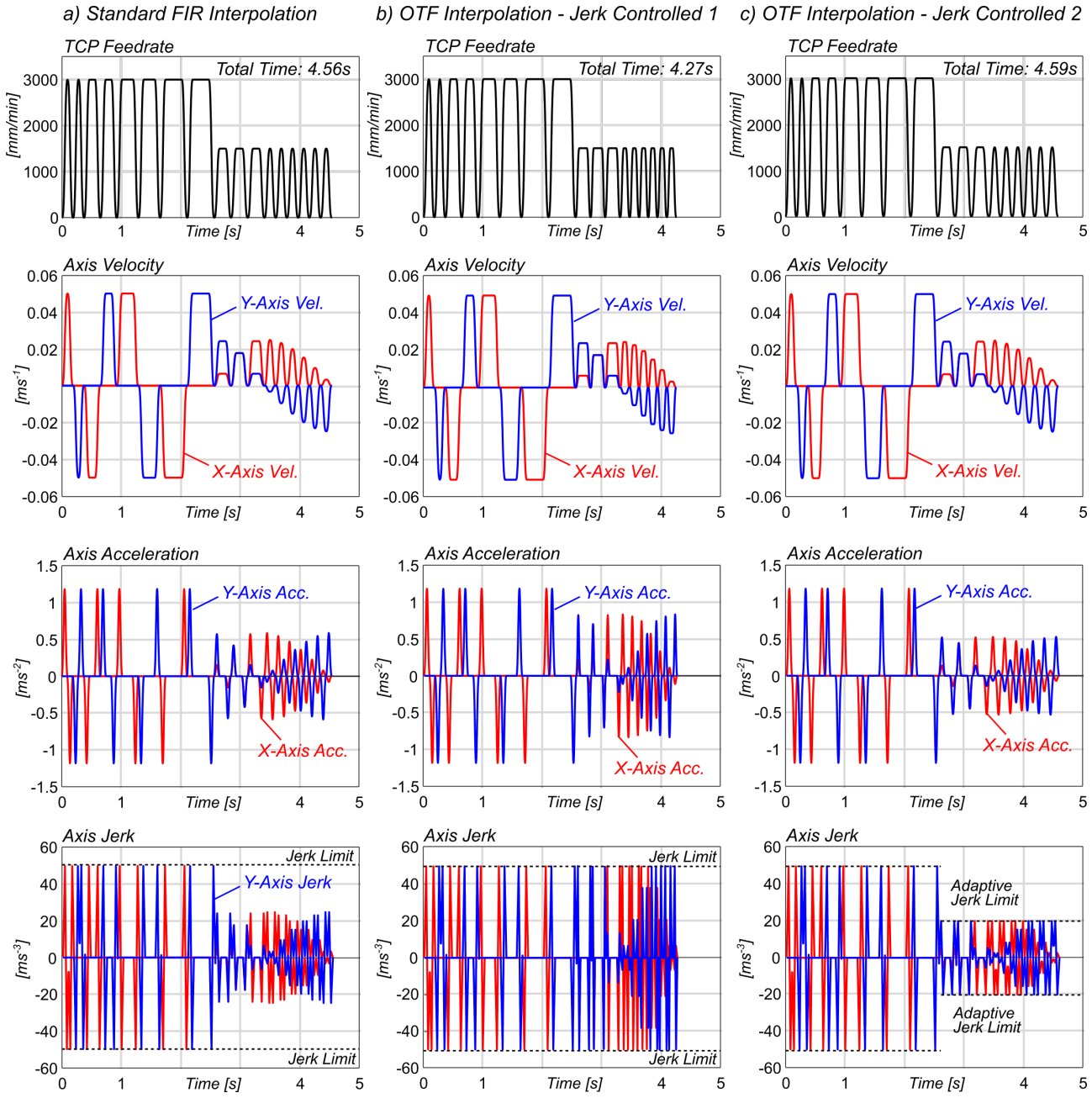

Figure 4.3: Kinematic profiles for P2P motion generated from a) standard FIR interpolation, b) OTF interpolation with constant jerk limit and c) OTF interpolation with adaptive jerk limit

4.2.3 OTF Trajectory Generation of 3-Axis Non-Stop Continuous Motion

In the previous section, it was shown P2P motion is achieved by end to end concatenation of the segmented smoothed position signals. Overlapping these signals during signal reconstruction blends the TCP cornering trajectory and modifies the cornering feedrates. The amount of overlap can be used to control the cornering feedrate and TCP blending error. An example overlap between two interpolated feedrate signals is shown in Fig.4.4. It can be seen the second TCP feedrate profile starts before the first feedrate profile reaches zero, resulting in an overlap of the two signals.

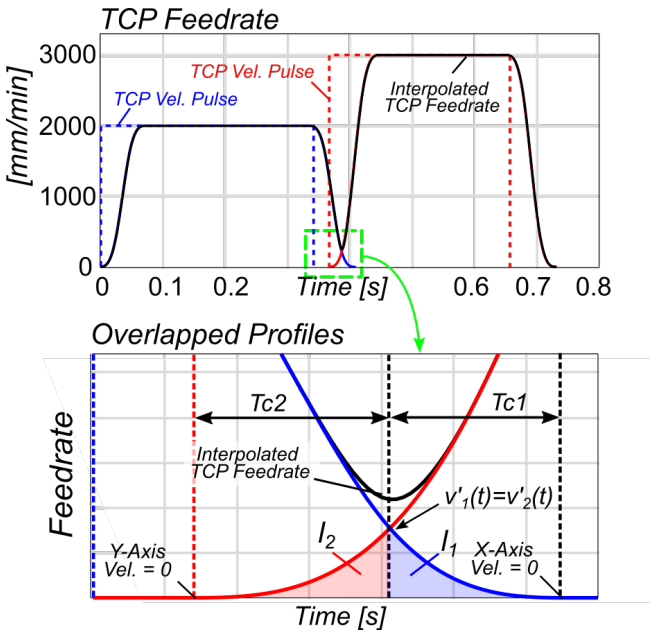

Figure 4.4: Overlap in interpolated segmented input signals

The proposed method for OTF interpolation is presented in Fig.4.5. First, the velocity pulses for each G01 command are generated using (4.1) and (4.2). The axis velocity pulses are smoothed using FIR-based interpolation (4.3). The smoothed segmented velocity command are integrated to yield the segmented smoothed position commands (4.6). The segmented signals are then reconstructed to yield the full interpolated reference signals.

As an illustrative example Fig.4.6 shows the kinematic profiles for a cornering motion with 2 different feedrates interpolated using OTF FIR interpolation. The main objective is to determine the overlap time between the segmented signals which satisfies both the TCP tolerance and kinematic limits. The overlap time $T_{o,TCP}$ is composed of two cornering times T_c:

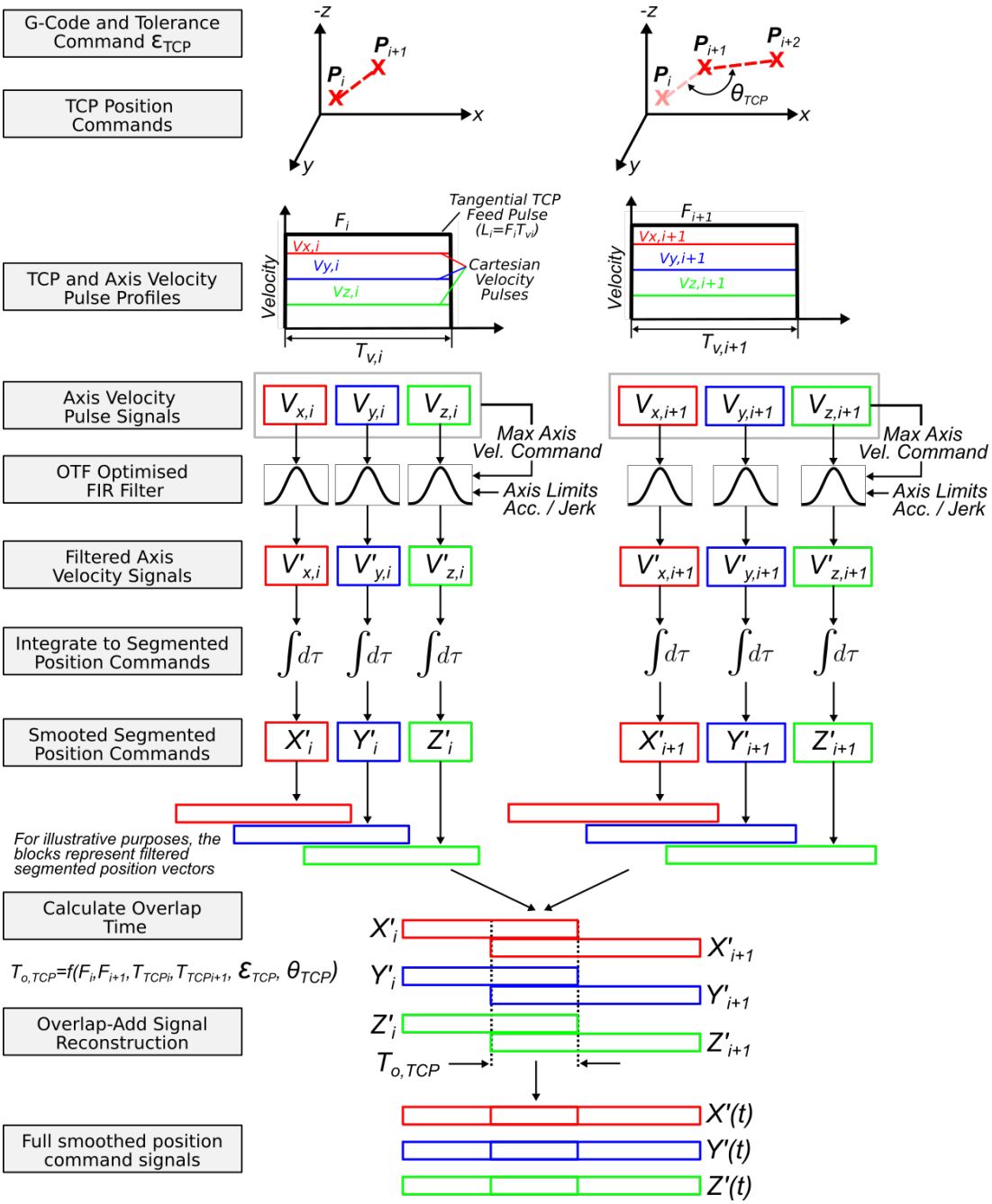

Figure 4.5: Overlap-add method of OTF interpolation of non-stop continuous 3-axis toolpaths

$$T_{o,TCP} = T_{c1} + T_{c2}, \tag{4.8}$$

where T_{c1} and T_{c2} represent the time taken from zero velocity to the intersection (crossover) of the smoothed velocity signals where $v_1' = v_2'$ as shown in Fig.4.4.

The total displacement s_{TCP} travelled along each segment at the time where $v_1' = v_2'$, is calculated from the piecewise analytical displacement equations (4.52) evaluated at $t = T_{c,i}$:

$$s_{TCP,i} = \frac{F_i T_{c,i}^4}{24 T_{TCP,i}^3}, \quad 0 \leq t \leq T_{TCP,i} \tag{4.9}$$

where $i = 1, 2$ for decelerating and accelerating segment respectively. The TCP blending error ε_{TCP} is calculated as the Euclidean distance from the CL position command and the closest TCP position on the interpolated trajectory, and using the displacements (4.9), is calculated as:

$$\varepsilon_{TCP}^2 = l_1^2 + l_2^2 + l_1 l_2 \cos\theta_{TCP}, \tag{4.10}$$

where l_1 and l_2 are the displacements along the linear motions and θ_{TCP} is the cornering angle between the two motions (see Fig.4.6). During constant feedrate, $F_1 = F_2$, the displacements are equal $l_1 = l_2$ and the interpolated trajectory is symmetrical about the corner. However, when $F_1 \neq F_2$, $l_1 \neq l_2$ and the interpolated trajectory is not symmetrical. The method to calculate T_{c1} and T_{c2} which satisfies (4.10) when $F_1 \neq F_2$ is presented as follows.

First, the TCP displacement equation (4.10) is solved for T_{c1} as a function of T_{c2}, resulting in:

$$T_{c1} = \left(F_1^{-1} T_2^{-3} T_1^3 \left(288 \sqrt{\frac{F_2^2 T_{c2}^8 \cos\theta_{TCP}^2}{82944} - \frac{F_2^2 T_{c2}^8}{82944} + \frac{T_2^6 \varepsilon_{TCP}^2}{144}} - F_2 T_{c2}^4 \cos\theta_{TCP} \right) \right)^{1/4} \tag{4.11}$$

Then, noting that the magnitude of axis velocities at the crossover point are equal $v'_{TCP,1} = v'_{TCP,2}$ (see Fig.4.4), the velocity equations yield an expression for T_{c2} as a function of T_{c1}:

$$\frac{F_1 T_{c1}^3}{6 T_1^3} = \frac{F_2 T_{c2}^3}{6 T_2^3} \rightarrow T_{c2} = \frac{F_1^{1/3} T_2 T_{c1}}{F_2^{1/3} T_1} \tag{4.12}$$

Solving (4.11) and (4.12) simultaneously leads to the following expressions for the two components of the overlap time:

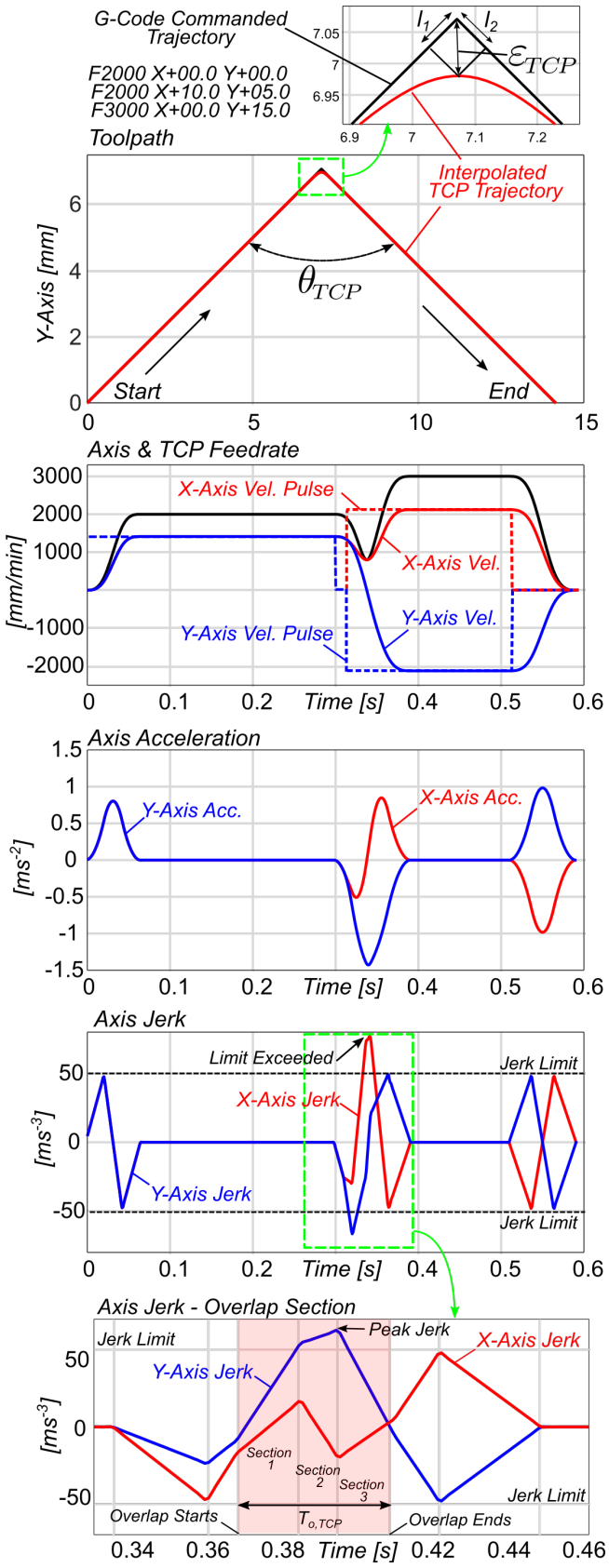

Figure 4.6: Illustrative toolpath and kinematic profiles showing segment overlap and exceeded jerk limit

$$T_{c1} = \left(\frac{576\, {F_2}^{2/3}\, {T_1}^{8}\, \varepsilon_{TCP}^2}{{F_1}^{8/3}\, {T_2}^{2} + {F_1}^{2}\, {F_2}^{2/3}\, {T_1}^{2} + 2\, {F_1}^{7/3}\, {F_2}^{1/3}\, T_1\, T_2\, \cos\theta_{TCP}} \right)^{1/8} \tag{4.13}$$

$$T_{c2} = \frac{{F_1}^{1/3}\, T_2}{{F_2}^{1/3}\, T_1} \left(\frac{576\, {F_2}^{2/3}\, {T_1}^{8}\, \varepsilon_{TCP}^2}{{F_1}^{8/3}\, {T_2}^{2} + {F_1}^{2}\, {F_2}^{2/3}\, {T_1}^{2} + 2\, {F_1}^{7/3}\, {F_2}^{1/3}\, T_1\, T_2\, \cos\theta_{TCP}} \right)^{1/8} \tag{4.14}$$

This is the first time the overlap time has been solved assuming 2 different feedrates. Previous methods of solving T_{c1} and T_{c2} for $F_1 \neq F_2$ overestimated the blending error by using the larger feedrate $F_{\max} = \max(F_1, F_2)$ and assumed equal displacements and a symmetrical trajectory. This method leads to an over conservative TCP error. Take the cornering motion in Fig.4.6 as an illustrative example. Assume the cornering angle changes from 0° to 180°. Figure 4.7 shows the TCP error for the OTF method of calculating the overlap time using (4.13) and (4.14) alongside the conservative case using $\max(F_1, F_2)$. It can be seen that using the OTF method demonstrates a TCP error closer to the user defined limit whereas the conservative method over constrains the corner trajectory ultimately leading to longer cycle times.

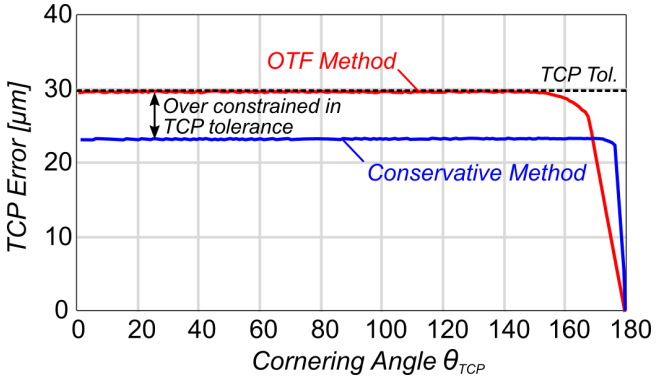

Figure 4.7: TCP error for cornering motion where $F_1 = 2000$ mm/min and $F_1 = 3000$ for 0 to 180°

Finally, the full interpolated position reference signals are reconstructed from the segmented smoothed displacement commands (4.6) using the OLA method governed by $T_{o,TCP}$:

$$\begin{bmatrix} x'(t) \\ y'(t) \\ z'(t) \end{bmatrix} = \begin{bmatrix} x'_i(t) \overset{T_{o,i}}{\bigcap} x'_{i+1}(t) \overset{T_{o,i+1}}{\bigcap} x'_{i+2}(t) \ldots\ x'_{N_{G01}}(t) \\ y'_i(t) \overset{T_{o,i}}{\bigcap} y'_{i+1}(t) \overset{T_{o,i+1}}{\bigcap} y'_{i+2}(t) \ldots\ y'_{N_{G01}}(t) \\ z'_i(t) \overset{T_{o,i}}{\bigcap} z'_{i+1}(t) \overset{T_{o,i+1}}{\bigcap} z'_{i+2}(t) \ldots\ z'_{N_{G01}}(t) \end{bmatrix}, \tag{4.15}$$

where $\overset{T_{o,i}}{\bigcap}$ represents the overlap of $T_{o,TCP,i}$ during signal reconstruction (see Fig.4.5).

This section presented a method to linearly interpolate each G01 command separately in order to maximise the kinematic performance throughout the toolpath. The overlap during OLA based signal reconstruction was analytically calculated to ensure TCP tolerance requirements were met. However, it is important to consider the effect of the overlap on both TCP and axis jerk. Within the overlapped regions the total axis jerk is a linear superposition of each overlapping jerk profile and this has the potential to exceed axis kinematic limits unless compensated as highlighted in the axis jerk profiles in Fig.4.6. The next section will address jerk control during non-stop continuous motion.

4.2.4 Jerk Control during Non-Stop Continuous Motion

As highlighted, the total axis jerk is a result of the linear superposition of the axis jerk profile for each overlapping segment. For P2P machining no overlap exists, therefore the jerk profile cannot exceed the maximum set jerk (as seen in Fig.4.3) as the FIR filter was specifically designed to maximise the jerk based on a single P2P motion using the local feedrate command (Eq.(4.5)). During non-stop continuous motion, however, the blended profiles can exceed the maximum jerk due to the overlapped sections, as can be seen in Fig.4.6. This peak jerk can be reduced to meet the jerk limit by several methods, the first option is to lower the feedrate, the second is to reduce the overlap, third, increase the FIR filter time constant, fourth, increase the order of the signal via an additional filter convolution and finally alter the velocity pulse edge profile. All methods increase the overall time but by differing amounts. The objective remains to minimise the machining cycle time whilst satisfying TCP tolerance and kinematic limits. Here we present two methods to control the jerk during continuous machining.

Jerk Control Method 1 - Modify Overlap T_o

The first method presented to control the jerk during blended cornering motions modifies the length of the segment overlap. Figure 4.6 shows a zoomed version of the jerk profile during the overlap during the cornering motion. The resultant (superimposed) jerk profiles during the overlapped sections are defined analytically by Eq.(4.16). The equation for the maximum jerk is derived and subsequently used to calculate the jerk optimised overlap time which reduces the peak jerk to the user set limit.

The first two sections of the overlap (as seen in the zoomed jerk profiles in Fig.4.6) can be

represented as:

$$j'(t) = \begin{cases} -\frac{F_2}{T_{TCP2}^3}\left(3\,T_{TCP1} - T_{o,TCP} + T_{v1} - t\right) - \frac{F_1}{T_{TCP1}^3}\left(3\,T_{TCP1} + 2\,T_{v1} - 2\,t\right), & \text{Section 1} \\ \frac{F_1}{T_{TCP1}^3}\left(3\,T_{TCP1} + T_{v1} - t\right) - \frac{F_2}{T_{TCP2}^3}\left(3\,T_{TCP1} - T_{o,TCP} + T_{v1} - t\right), & \text{Section 2} \end{cases} \tag{4.16}$$

where $t = 0$ represents the start of the first motion $P_1 = [0, 0, 0]$ in Fig.4.6. Evaluating (4.16) at the time of peak jerk results in:

$$j'_{\text{peak}} = \frac{F_2\,{T_{TCP1}}^3 - F\,{T_{TCP2}}^3 + F_1\,T_{o,TCP}\,{T_{TCP2}}^2}{{T_{TCP1}}^3\,{T_{TCP2}}^2}. \tag{4.17}$$

Rearranging (4.17) and using the jerk limit $J_{\max}$ results in the expression for the minimum overlap $T_{o,TCP}$ which satisfies the jerk constraint during the overlap:

$$T_{o,TCP,\text{Jmax}} = \frac{J_{max}\,{T_{TCP1}}^3\,{T_{TCP2}}^2 - F_2\,{T_{TCP1}}^3 + F_1\,{T_{TCP2}}^3}{F\,{T_{TCP2}}^2}, \tag{4.18}$$

However, driving the jerk to the limit does not guarantee faster cycle times as it can force the motion to P2P in order to satisfy TCP tolerance. At this stage, $T_{o,TCP}$ has been calculated satisfying TCP tolerance ε_{TCP} and maximum jerk $J_{\max}$ separately. The lower value is selected to ensure both constraints are met, such that $T_{o,TCP} = \min[T_{o,TCP,TOL}, T_{o,TCP,\text{Jmax}}]$. However, this reduction in overlap can significantly increase the overall time of the blend resulting in more accurate tolerances but longer machining cycle times. Another consideration is to alter the velocity pulse profiles.

Jerk Control Method 2 - Modify Velocity Pulse Shape

The key principle in FIR-based interpolation of NC toolpaths has been to smooth square velocity pulses [21, 25]. However, changing the profile of the square velocity pulse, such that it is stepped for example, alters the kinematic profile of the smoothed interpolated output signal. Here we present a method which inserts a single step at either the rising or falling edges of the velocity pulse in order to change the resulting interpolated jerk profile and reduce the maximum peak jerk.

Figure 4.8 shows a velocity pulse with a square rising edge and stepped falling edge along with the associated interpolated kinematic profiles. The stepped edge has a height of $F/2$ and a width of T_{TCP}. As the area of pulse is equal to L, the new total length of the pulse is $T_v + T_{TCP}/2$. Therefore each stepped pulse edge increases the overall pulse length by $T_{TCP}/2$

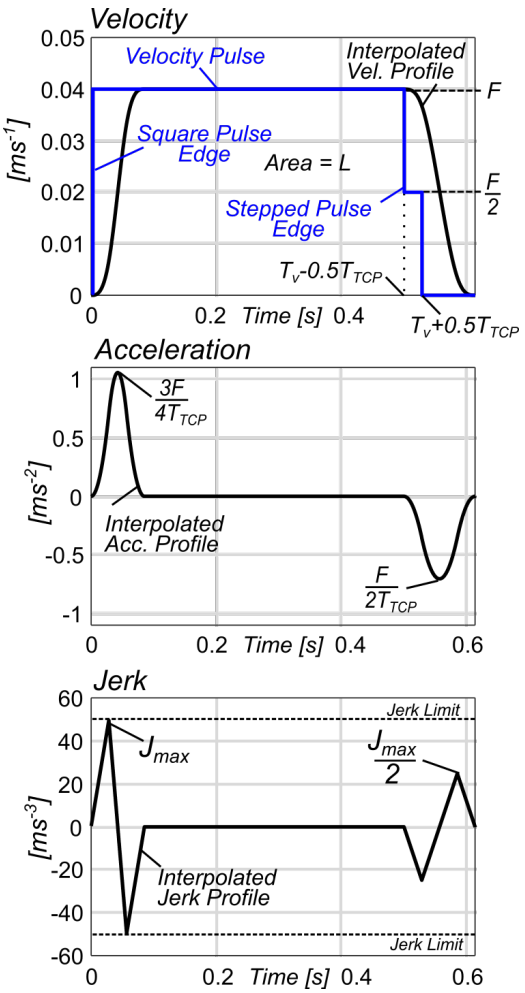

Figure 4.8: Interpolated kinematic profiles from a square and stepped edge velocity pulse

compared to the standard square edge. Despite this however, as will be shown, time savings are made within the newly calculated overlap. Returning to the interpolated kinematic profiles, Fig.4.8 shows peak acceleration of the stepped edge is reduced by 33% when compared to the square edge. Similarly, when viewing the jerk profiles the square edge results in the peak jerk equal to $J_{\max}$ as expected from Eq.(4.5), however, the peak jerk drops to $J_{\max}/2$ for the stepped edge, a 50% reduction. The considerable reduction in peak jerk for the stepped edge enables the interpolated segments to increase the overlap. As the kinematic profiles have been modified, a new T_o must be calculated.

Similar to the standard square pulse, the analytical profiles of the interpolated stepped pulse are used to calculate the overlap time which meets the user set TCP tolerance. The analytical equations for the stepped edge interpolated velocity pulse are as follows (first two sections only for brevity):

$$s'_{TCP}(t) = \begin{cases} \frac{F}{48T_{TCP}^3}t^4, & 0 \leq t < T_{TCP} \\ -\frac{F}{48{T_{TCP}}^3}\left(12T_{TCP}^4 + 8T_{TCP}^3 t + 12T_{TCP}^2 t^2 - 8T_{TCP}t^3 + t^4\right), & T_{TCP} \leq t < 2T_{TCP} \end{cases} \tag{4.19}$$

$$v'(t) = \begin{cases} \frac{F}{12T_{TCP}^3}t^3, & 0 \leq t < T_{TCP} \\ \frac{F}{6} - \frac{F}{12T_{TCP}^3}\left(6T_{TCP}^2 t - 6T_{TCP}t^2 + t^3\right), & T_{TCP} \leq t < 2T_{TCP} \end{cases} \tag{4.20}$$

$$a'(t) = \begin{cases} \frac{F}{4T_{TCP}^3}t^2, & 0 \leq t < T_{TCP} \\ \frac{-F}{4T_{TCP}^3}\left(2T_{TCP}^2 - 4T_{TCP}t + t^2\right), & T_{TCP} \leq t < 2T_{TCP} \end{cases} \tag{4.21}$$

$$j'(t) = \begin{cases} \frac{F}{2T_{TCP}^3}t, & 0 \leq t < T_{TCP} \\ \frac{F}{2T_{TCP}^3}\left(2T_{TCP} - t\right). & T_{TCP} \leq t < 2T_{TCP} \end{cases} \tag{4.22}$$

The stepped edge displacement (4.19) and velocity equations (4.20) are used to calculate the stepped edge based overlap $T_{o,\text{step}}$ using the the same method as presented in section 4.2.3. The stepped edge based overlap is calculated as $T_{o,\text{step}} = T_{c1,\text{step}} + T_{c2,\text{step}}$ where:

$$T_{c1,\text{step}} = \left(\frac{24^2\, {F_2}^{2/3}\, {T_{TCP1}}^8\, \varepsilon_{TCP}^2}{{F_1}^{8/3}\, {T_{TCP2}}^2 + {F_1}^2\, {F_2}^{2/3}\, {T_{TCP1}}^2 + 2\, {F_1}^{7/3}\, {F_2}^{1/3}\, T_{TCP1}\, T_{TCP2}\, \cos\theta_{TCP}}\right)^{1/8}, \tag{4.23}$$

$$T_{c2,\text{step}} = \frac{{F_1}^{1/3}\, T_2}{{F_2}^{1/3}\, T_1} \cdot T_{c1,\text{step}} \tag{4.24}$$

Thus, when using the proposed OTF method, if the axis jerk is exceeded (calculated online using Eq.(4.17)), then the velocity pulse edge is modified to a stepped edge and the new overlap time calculated using (4.23) and (4.24).

Illustrative 3-Axis OTF Case Studies

In this section we present 3 case studies compare the OTF interpolation method to the standard FIR interpolation method. The objective is to show how the standard OTF method does not satisfy jerk alone and that an element of jerk control must be considered. The lock toolpath and associated NC code in Fig.4.2 is once again used. Figure 4.9 shows the kinematic profiles and TCP error for the 4 cases. The case study is demonstrated for a semi-finishing case such that the jerk limit is 30 ms^{-3} and the TCP tolerance is 30μms.

The first case (Fig.4.9a) shows the standard FIR interpolation method in which the FIR filter time constant does not change throughout the toolpath. As such, the time constant is selected

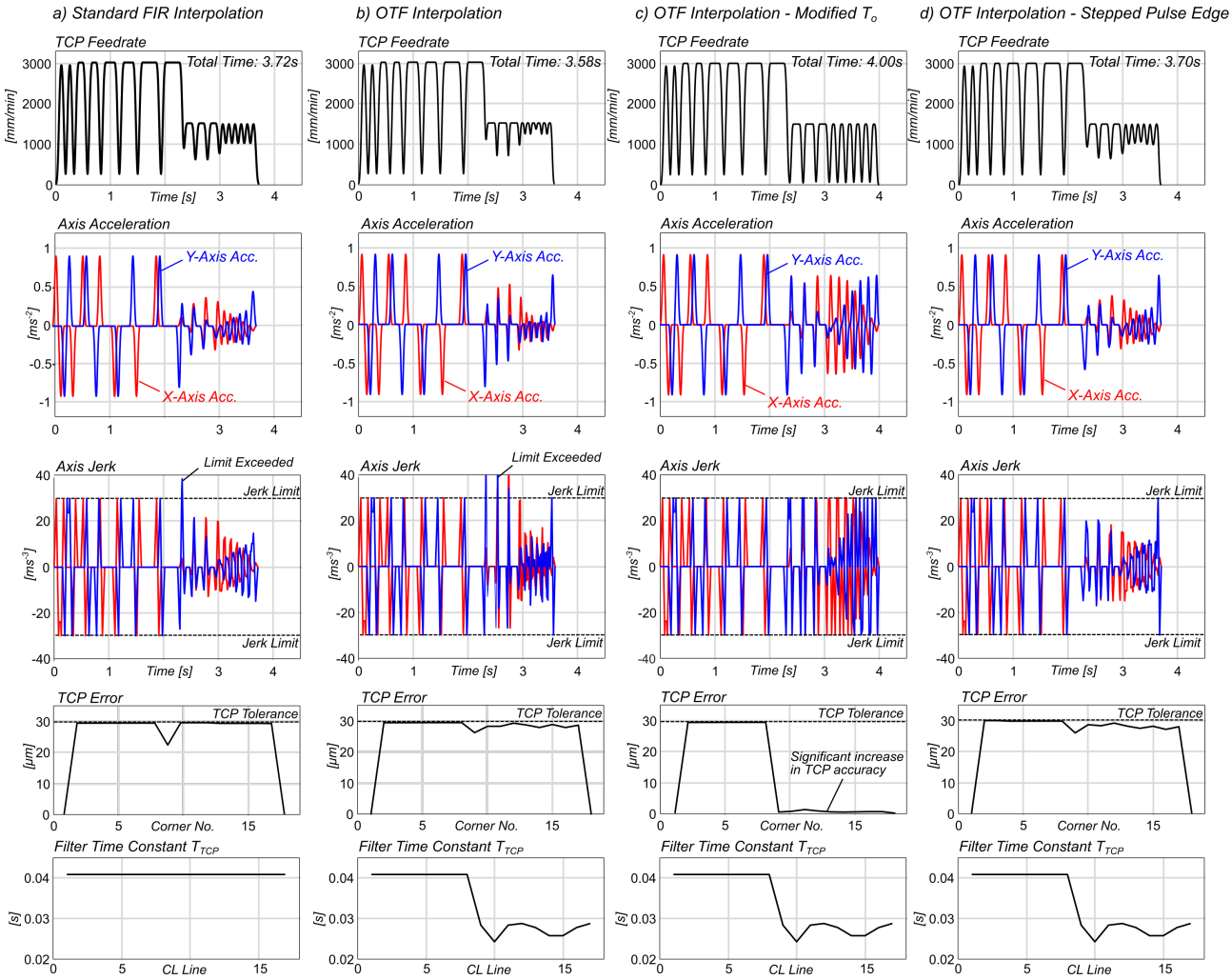

Figure 4.9: Kinematic profiles for continuous non-stop motion generated from a) OTF interpolation, b) OTF interpolation with jerk limited T_o and c) OTF interpolation with step edged velocity pulses

based on the maximum feedrate in the part program, i.e. it it optimised for 3000mm/min in this example. The total machining time is 3.72s. It can be seen from the kinematic profiles that the jerk is exceeded when the toolpath transitions to the arc trajectory. The TCP error is satisfied throughout the toolpath but over compensates during the change in feedrate command when $F_i \neq F_{i+1}$ as described in section 4.2.3.

The second case (Fig.4.9b) uses the OTF interpolated method presented in section 4.2.3. The overlap $T_{o,TCP}$ is calculated from (4.13) and (4.14). It can be seen from the kinematic profiles that the jerk exceeds the limit around the arc section of the toolpath (38.6 and 41.8 ms^{-3} for the X and Y axis respectively). Despite this, the TCP tolerance is within limits throughout the toolpath and the total time is 3.58s. Therefore the OTF method is 3.8% faster than the standard FIR method in this example. In application, the actual time savings are highly dependent on toolpath geometry, jerk limits and TCP tolerance.

The third case (Fig.4.9c) considers the jerk limit. The OTF interpolation method from 4.2.3 is again used in conjunction with the $T_{o,TCP,\text{Jmax}}$ from (4.18). For any segments which exceed the jerk limit, the lower value of $\min[T_{o,TCP,TOL}, T_{o,TCP,\text{Jmax}}]$ is used. Inspecting the jerk profile, the method drives the overlap time to almost 0, this effectively leads to near P2P motion as can be seen in the feedrate profile and also in the significant increase in TCP accuracy. Despite, the jerk limit and TCP tolerance being satisfied, this method leads to longer machining cycle times (total time = 4.00s) and higher precision than is required.

Finally, the fourth case (Fig.4.9d) introduces the OTF method using the stepped velocity pulses. If the axis jerk constraint is exceeded then $T_{o,\text{step}}$ from (4.23) and (4.24) is applied. The axis jerk limit is not exceeded throughout the toolpath and the accuracy remains within tolerance. The total time is 3.70s. Compared to the third case (which satisfies all constraints) this is a 7.5% reduction in cycle time.

This section has introduced OTF interpolation for both P2P and non-stop continuous motion 3-axis toolpaths. The next section will extend OTF interpolation to 5-axis machine tool motion.

4.3 5-Axis OTF Trajectory Generation

On-The-Fly NC interpolation of 3-axis toolpaths was presented in section 4.2. The method is now extended to 5-axis toolpaths. The method of interpolating and transforming 5-axis tool position and orientation from G01 commands to feed drive commands in the MCS was presented in chapter 3. The general method is summarised in section 4.3.1 for completeness.

4.3.1 5-Axis ORI Interpolation

The interpolation of the TCP in 5-axis follows the same method as presented in the 3-axis case as this is conducted in the workpiece coordinate system (WCS). As tool orientation commands are interpolated in the machine coordinate system (MCS), inverse kinematic transformations (IKTs) from the WCS to the MCS are applied to the initial tool orientation $\mathbf{O}_i = [O_{i,i}, O_{i,j}, O_{i,k}]^T$ and final $\boldsymbol{O_{i+1}}$ command of the G01 segment. The initial and final rotary axis positions are computed as:

$$\begin{bmatrix} A_i \\ C_i \end{bmatrix} = IKT_{ROT}\,(O_{i,i}, O_{j,i}, O_{k,i}), \tag{4.25a}$$

$$\begin{bmatrix} A_{i+1} \\ C_{i+1} \end{bmatrix} = IKT_{ROT}\,(O_{i,i+1}, O_{j,i+1}, O_{k,i+1}), \tag{4.25b}$$

where IKT_{ROT} is the IKT for the rotary axis positions (A,C).

The rotary axis velocity commands are calculated in the MCS. First, the rotary feedrates $(F_A(t), F_C(t))$ are calculated as:

$$\begin{bmatrix} F_A(t) \\ F_C(t) \end{bmatrix} = \begin{bmatrix} A_{i+1} - A_i \\ C_{i+1} - C_i \end{bmatrix} / T_{v,i}. \quad 0 < t \leq T_{v,i} \tag{4.26}$$

The segmented rotary axis velocity pulses are then linearly interpolated in the MCS by filtering with a low-pass filter $G_{ROT,i}(s)$:

$$\begin{bmatrix} v'_{A.i}(t) \\ v'_{C,i}(t) \end{bmatrix} = \begin{bmatrix} F_{A,i}(t) \\ F_{c,i}(t) \end{bmatrix} * m_{ROT,i}(t), \tag{4.27}$$

where $m_{ROT,i}\,(t)$ is the impulse response of $G_{ROT,i}(s)$, the rotary FIR filter transfer function. The filter $G_{ROT}(s)$ is designed with the same structure and order as the filter $G_{TCP}(s)$ but its time constant T_{ROT} can be selected such that:

$$T_{ROT} = \frac{3\Delta F_{ROT,i}}{4A_{ROT}^{\max}} \tag{4.28}$$

where $\Delta F_{ROT,i} = \max(F_{A,i}(t), F_{c,i}(t))$. During 5-axis OTF interpolation to ensure synchronisation between the TCP and ORI motions the time constant is selected for each individual interpolation operation as:

$$T_i = \max\left\{\frac{3\Delta F_{ROT,i}}{4A_{ROT}^{\max}}, \frac{3\Delta F_i}{4A_{TCP}^{\max}}, \sqrt{\frac{\Delta F_i}{J_{TCP}^{\max}}}\right\}. \tag{4.29}$$

Following FIR interpolation, the filtered rotary velocity signals are integrated to generate the segmented rotary axes position commands:

$$\begin{bmatrix} A'_i(t) \\ C'_i(t) \end{bmatrix} = \begin{bmatrix} A_i \\ C_i \end{bmatrix} + \int_0^t \begin{bmatrix} v'_{A.i}(t) \\ v'_{C,i}(t) \end{bmatrix} d\tau. \tag{4.30}$$

The segmented rotary axis position signals are reconstructed using the OLA method for either P2P or continuous machining.

4.3.2 OTF Trajectory Generation of 5-Axis P2P Motion

For 5-axis P2P toolpaths, the same signal reconstruction method as presented in section 4.2.2 is used. The OLA method of signal reconstruction concatenates the segmented smoothed rotary position command signals and ensures an instantaneous stop in between each commanded rotary axis position. The full smoothed P2P rotary axis positions signals $A'(t)$ and $C'(t)$ are constructed as:

$$\begin{bmatrix} A'(t) \\ C'(t) \end{bmatrix} = \begin{bmatrix} A'_i(t), A'_{i+1}(t), ..., A'_{N_{G01}}(t) \\ C'_i(t), C'_{i+1}(t), ..., C'_{N_{G01}}(t) \end{bmatrix}. \tag{4.31}$$

The interpolated TCP position commands in the WCS are generated using the method in section 4.2. Finally, using the interpolated TCP position commands $[x'(t), y'(t), z'(t)]$ and interpolated rotary axis position commands $[A'(t), C'(t)]$, the Cartesian axis positions commands $[X', Y', Z']$ in the MCS can be generated by applying the TCP inverse kinematics:

$$\begin{bmatrix} X'(t) \\ Y'(t) \\ Z'(t) \end{bmatrix} = IKT_{TCP}\left(x'(t), y'(t), z'(t), A'(t), C'(t)\right), \tag{4.32}$$

where IKT_{TCP} is the TCP's inverse kinematics. The newly generated reference signals in (4.31) and (4.32) are used to command the machine tool feed drives.

The following section will show how the overlap between successive segmented signals in the rotary MCS can confine the ORI interpolation errors in the WCS.

4.3.3 OTF Trajectory Generation of 5-Axis Non-Stop Continuous Motion

In this section the OTF method of NC interpolation is applied 5-axis machining. The tool orientation tolerance in the WCS is satisfied through controlling the rotary axis blending errors in the MCS. Similar to presented method of controlling 3-axis continuous motion, the 5-axis method uses the overlap of the smoothed segmented input signals to control the displacement and velocity profiles in the blends. The objective therefore is to calculate the rotary axis overlap $T_{o,ROT}$ for each segment which satisfies both the tool orientation tolerance and rotary drive kinematic limits. The rotary axis overlap is constructed from the two rotary corner times $T_{o,ROT=}T_{c1,ROT}$ and $T_{c2,ROT}$. The following section will derive $T_{c1,ROT}$ and $T_{c2,ROT}$.
The rotary axis overlap $T_{o,ROT}$ is analytically calculated using the piecewise kinematic equations for the interpolated segments. The piecewise analytical kinematic equations for rotary displacement are as follows:

$$s_\theta(t) = \begin{cases} \frac{F_\theta t^4}{24{T_{ROT}}^3}, & 0 \le t \le T_{ROT} \\ \frac{F_\theta}{24{T_{ROT}}^3}\left(-3{T_{ROT}}^4 + 12{T_{ROT}}^3 t - 18{T_{ROT}}^2 t^2 + 12 T_{ROT} t^3 - 2t^4\right), & T_{ROT} \le t \le 2T_{ROT} \\ \frac{F_\theta}{24{T_{ROT}}^3}\left(45{T_{ROT}}^4 - 84{T_{ROT}}^3 t + 54{T_{ROT}}^2 t^2 - 12 T_{ROT} t^3 + t^4\right), & 2T_{ROT} \le t \le 3T_{ROT} \\ F_\theta t - \frac{3}{2} F_\theta T_{ROT}, & 3T_{ROT} \le t \le T_{v,ROT} \end{cases} \tag{4.33}$$

where subscript $\theta = [A, C]$ and T_{ROT} is the rotary FIR filter time constant. For precision machining tolerances, the rotary axis displacements at $T_{c1,ROT}$ and $T_{c2,ROT}$ are calculated within the first kinematic section $0 \le t \le T_{ROT}$.

$$s_{\theta i}(t) = \frac{F_\theta T_{ci,ROT}^4}{24{T_{ROT,i}}^3}, \quad 0 \le t \le T_{ROT,i} \tag{4.34}$$

where subscript $i = [1, 2]$ for the decelerating and accelerating segment respectively, i.e. first and second segment of a CL-junction. The total rotary displacement errors, $[\delta A, \delta C]$, for each axis are calculated from the displacement (using (4.34)) of the accelerating section minus the decelerating section as:

$$\delta A = s_{A1} - s_{A2} = \frac{F_{A1} T_{c1,ROT}^4}{24 T_{ROT1}^3} - \frac{F_{A2} T_{c2,ROT}^4}{24 T_{ROT2}^3}, \tag{4.35}$$

$$\delta C = s_{C1} - s_{C2} = \frac{F_{C1} T_{c1,ROT}^4}{24 T_{ROT1}^3} - \frac{F_{C2} T_{c2,ROT}^4}{24 T_{ROT2}^3}. \tag{4.36}$$

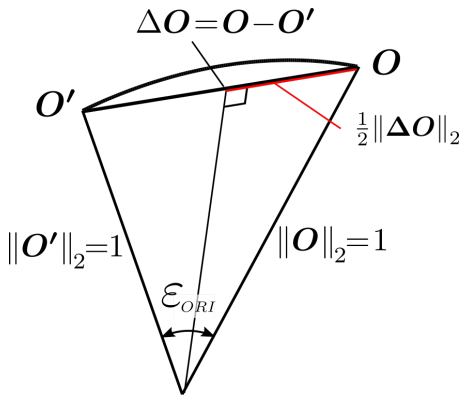

Figure 4.10: Interpolated tool orientation error in spherical coordinates (WCS)

The rotary displacement error vector $\Delta\boldsymbol{\theta}$ is formed by combining (4.35) and (4.36) into vector form:

$$\Delta\boldsymbol{\theta} = \begin{bmatrix} \delta A \\ \delta C \end{bmatrix} = \frac{T^4_{c1,ROT}}{24T^3_{ROT1}} \begin{bmatrix} F_{A1} \\ F_{C1} \end{bmatrix} - \frac{T^4_{c2,ROT}}{24T^3_{ROT2}} \begin{bmatrix} F_{A2} \\ F_{C2} \end{bmatrix}, \tag{4.37}$$

The rotary displacement errors defined in (4.37) will now be related to the deviation in the interpolated tool orientation vector in the WCS (spherical coordinates) to the commanded tool orientation vector through the rotary Jacobian. The geometrical relationship between the interpolated and commanded tool orientation vectors is presented in Fig.4.10. The interpolated tool orientation angular deviations are defined by ε_{ORI}:

$$\frac{1}{2}\varepsilon_{ORI} = \sin^{-1}\left(\frac{\frac{1}{2}\|\Delta\mathbf{O}\|_2}{\|\mathbf{O}'\|_2}\right) \tag{4.38}$$

The ORI error vector $\Delta\mathbf{O}$ is approximated from the rotary displacement errors $\Delta\boldsymbol{\theta} = [\Delta A, \Delta C]^T$ through the rotary Jacobian matrix $\mathbf{J_{ROT}}$. Both ε_{ORI} and $\Delta\boldsymbol{\theta}$ are sufficiently small such that both small angle and Jacobian approximation are valid, resulting in:

$$\Delta\boldsymbol{O} \approx \mathbf{J_{ROT}}\Delta\boldsymbol{\theta} \tag{4.39}$$

where,

$$\mathbf{J_{ROT}} = \begin{bmatrix} \frac{\partial O_i}{\partial A} & \frac{\partial O_i}{\partial C} \\ \frac{\partial O_j}{\partial A} & \frac{\partial O_j}{\partial C} \\ \frac{\partial O_k}{\partial A} & \frac{\partial O_k}{\partial C} \end{bmatrix}. \tag{4.40}$$

Using (4.38) and (4.39), the tool orientation tolerance $\varepsilon_{ORI,TOL}$ is guaranteed when:

$$\|\mathbf{J_{ROT}}\Delta\boldsymbol{\theta}\|_2 \leq \sin\varepsilon_{ORI,TOL} \tag{4.41}$$

Inserting the full rotary axis displacement errors $\Delta\boldsymbol{\theta}$ from (4.37),

$$\mathbf{J_{ROT}}\Delta\boldsymbol{\theta} \approx \frac{T_{c1,ROT}^4}{24T_{ROT1}^3}\mathbf{J_{ROT}}\begin{bmatrix} F_{A1} \\ F_{C1} \end{bmatrix} - \frac{T_{c2,ROT}^4}{24T_{ROT2}^3}\mathbf{J_{ROT}}\begin{bmatrix} F_{A2} \\ F_{C2} \end{bmatrix} \tag{4.42}$$

and using the rotary Jacobian to approximate the tool angular velocities from the rotary axis velocities:

$$\left\| \mathbf{J_{ROT}}\begin{bmatrix} F_A \\ F_C \end{bmatrix} \right\|_2 = \left\| \begin{matrix} \omega_i \\ \omega_j \\ \omega_k \end{matrix} \right\|_2 = \omega_{ORI} \tag{4.43}$$

and taking the Euclidian norm (using the reverse triangle inequality for matrix norms $|x-y| \geq ||x|-|y||$), results in the following inequality relating the MCS commands to the WCS tool orientation tolerance:

$$\frac{\omega_1}{24T_{ROT1}^3}T_{c1,ROT}^4 - \frac{\omega_2}{24T_{ROT2}^3}T_{c2,ROT}^4 - \sin\left(\varepsilon_{ORI}\right) \leq 0 \tag{4.44}$$

which yields conservative values of $T_{c1,ROT}$ and $T_{c2,ROT}$ to satisfy the ORI tolerance ε_{ORI}.
Next, the rotary velocities are used to generate a second set of equations for calculating $T_{c1,ROT}$ and $T_{c2,ROT}$. The piecewise rotary axis velocities are analytically defined as follows:

$$v_\theta(t) = \begin{cases} \frac{F_\theta t^3}{6{T_{ROT}}^3}, & 0 \leq t < T_{ROT} \\ \frac{F_\theta}{2\,{T_{ROT}}^3}\left({T_{ROT}}^3 - 3\,{T_{ROT}}^2\,t + 3\,T_{ROT}\,t^2 - \frac{2\,t^3}{3}\right) & T_{ROT} \leq t < 2T_{ROT} \\ -\frac{F_\theta}{6\,{T_{ROT}}^3}\left(21\,{T_{ROT}}^3 - 27\,{T_{ROT}}^2\,t + 9\,T_{ROT}\,t^2 - t^3\right) & 2T_{ROT} \leq t < 3T_{ROT} \\ F_\theta, & 3T_{ROT} \leq t < T_v \end{cases} \tag{4.45}$$

where subscript $\theta = [A, C]$. The magnitude of the axis velocities are equal when the interpolated axis velocity profiles crossover, where

$$\begin{bmatrix} |v'_{A1}| \\ |v'_{C1}| \end{bmatrix} = \begin{bmatrix} |v'_{A2}| \\ |v'_{C1}| \end{bmatrix} \tag{4.46}$$

The rotary crossover velocities are determined within the first kinematic sections $0 \leq t < T_{ROT}$, resulting in the following expression:

$$\frac{T_{c1,ROT}^3}{24T_{ROT1}^3}\begin{bmatrix} F_{A1} \\ F_{C1} \end{bmatrix} = \frac{T_{c2,ROT}^3}{24T_{ROT2}^3}\begin{bmatrix} F_{A2} \\ F_{C2} \end{bmatrix} \tag{4.47}$$

Multiplying both sides by the rotary Jacobian (4.43) and taking the Euclidean norm converts the MCS based rotary velocity equations into a single WCS based angular velocity equation:

$$\frac{T_{c2,ROT}^3 \omega_2}{24 T_{ROT2}^3} - \frac{T_{c1,ROT}^3 \omega_1}{24 T_{ROT1}^3} = 0 \tag{4.48}$$

Equations (4.44) and (4.48) have now been derived from the displacement and velocity equations respectively which can be simultaneously solved to yield real positive solutions for $T_{c1,ROT}$ and $T_{c2,ROT}$:

$$T_{c1,ROT} = \left(\frac{24\, {T_{ROT1}}^4\, {\omega_2}^{1/3}\, \sin(\varepsilon_{ORI})}{T_{ROT2}\, {\omega_1}^{4/3} + T_{ROT1}\, \omega_1\, {\omega_2}^{1/3}} \right)^{1/4} \tag{4.49}$$

$$T_{c2,ROT} = \left(\frac{24\, {T_{ROT2}}^4\, {\omega_1}^{1/3}\, \sin(\varepsilon_{ORI})}{T_{ROT1}\, {\omega_2}^{4/3} + T_{ROT2}\, {\omega_1}^{1/3}\, \omega_2} \right)^{1/4} \tag{4.50}$$

Finally the rotary axis overlap time $T_{o,ROT=}T_{c1,ROT} + T_{c2,ROT}$ can be calculated. In order to maintain synchronisation the same overlap time is selected for both the Cartesian and Rotary input signals. The lowest value is selected such that:

$$T_o = \min[T_{o,TCP}, T_{o,ROT}]. \tag{4.51}$$

This section has extended OTF interpolation to continuous 5-axis machining. The following sections benchmarks the OTF method against the standard FIR interpolation method.

4.4 Illustrative Examples

In this section the proposed 5-axis interpolation method is compared against the standard FIR interpolation method (with a fixed time constant) for both P2P and continuous machining. To demonstrate the generality of the method in terms of machine configuration, the OTF interpolation method is applied to a non-orthogonal machine configuration. The benchmark machine is the DMU eVo 40 5-axis non-orthogonal machine tool as shown in Fig.2.5. The forward and inverse kinematics for the non-orthogonal configuration are derived and presented in Appendix A.

The star shaped toolpath [25] shown in Fig.4.11 is commanded at 3000mm/min and the associated TCP position and orientation commands are given in Table.4.1 in Appendix 4B. The maximum Cartesian axis acceleration is 10ms^{-2} and the TCP jerk limit is set to 50ms^{-3}. To demonstrate the limitations of the rotary axes, the rotary axes acceleration limit is defined as $450°s^{-2}$. The star shaped toolpath is interpolated with the OTF FIR method and the standard FIR method. First, the P2P case is presented.

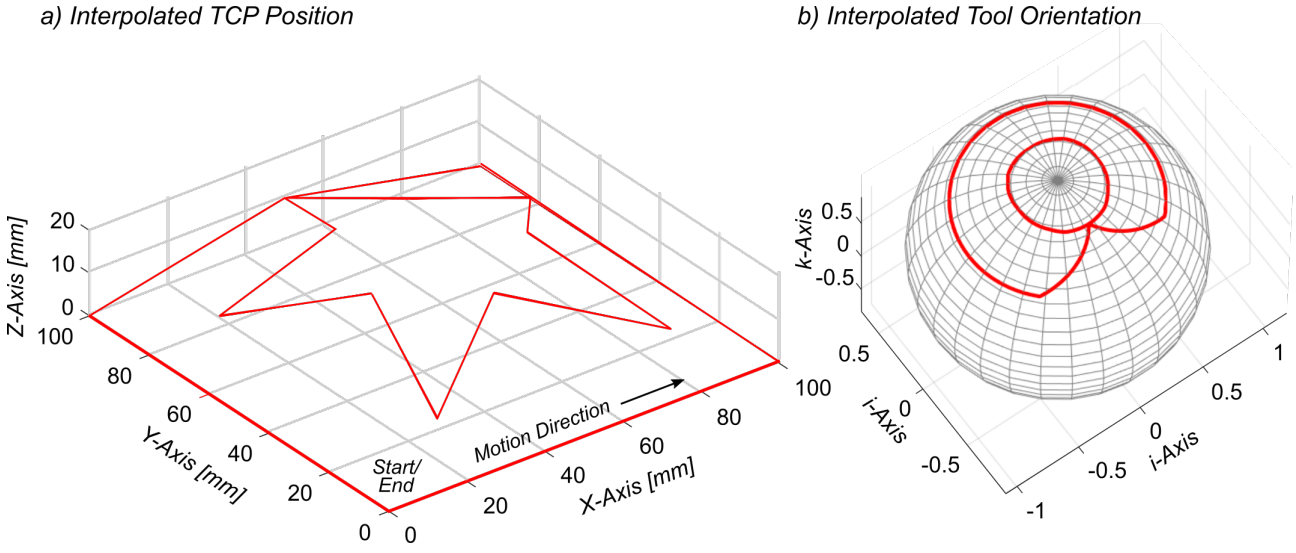

Figure 4.11: Interpolated TCP position and orientation for 5-axis star toolpath

4.4.1 P2P 5-Axis Motion

The interpolated kinematic profiles for the P2P case are shown in Fig.4.12. The results show the total machining cycle times are 20.06s and 18.66s for the standard FIR and OTF FIR interpolation methods respectively, demonstrating a 6.98% reduction in cycle time for the proposed OTF FIR method. This is attributed to the gain in kinematic performance during cornering motions.

In terms of the kinematic limits, the Cartesian acceleration and jerk limits are both satisfied (4.12b and 4.12c respectively). Noting, that satisfying the TCP acceleration limit also guarantees machine Cartesian kinematic limits. In 5-axis machining this is not a difficult constraint to meet as it is the performance of the rotary axes which is the limiting constraint.

The FIR filter time constants for the two cases are shown in Fig.4.12d. For the standard FIR case, the maximum rotary velocity during the toolpath combined with the rotary acceleration limit determines the FIR filter time constant to be $T_{ROT} = 0.113s$ using (4.28), which does not change. As can be seen in Fig.4.12d, when using the proposed OTF FIR method, the FIR filter time constant changes for each G-code command to maximise the kinematic performance of the machine tool feed drives - essentially increasing the magnitude of acceleration by reducing the FIR filter time constant. This is seen clearly by inspecting the rotary axes acceleration profiles in Fig.4.12g and 4.12h. The results show the standard FIR method only reaches the rotary acceleration limit once as opposed to the OTF method which maximises the response and reaches the limit for each individual motion.

The results from the P2P interpolation show that by using an adaptive FIR filter time constant significant reductions in machining cycle times are feasible.

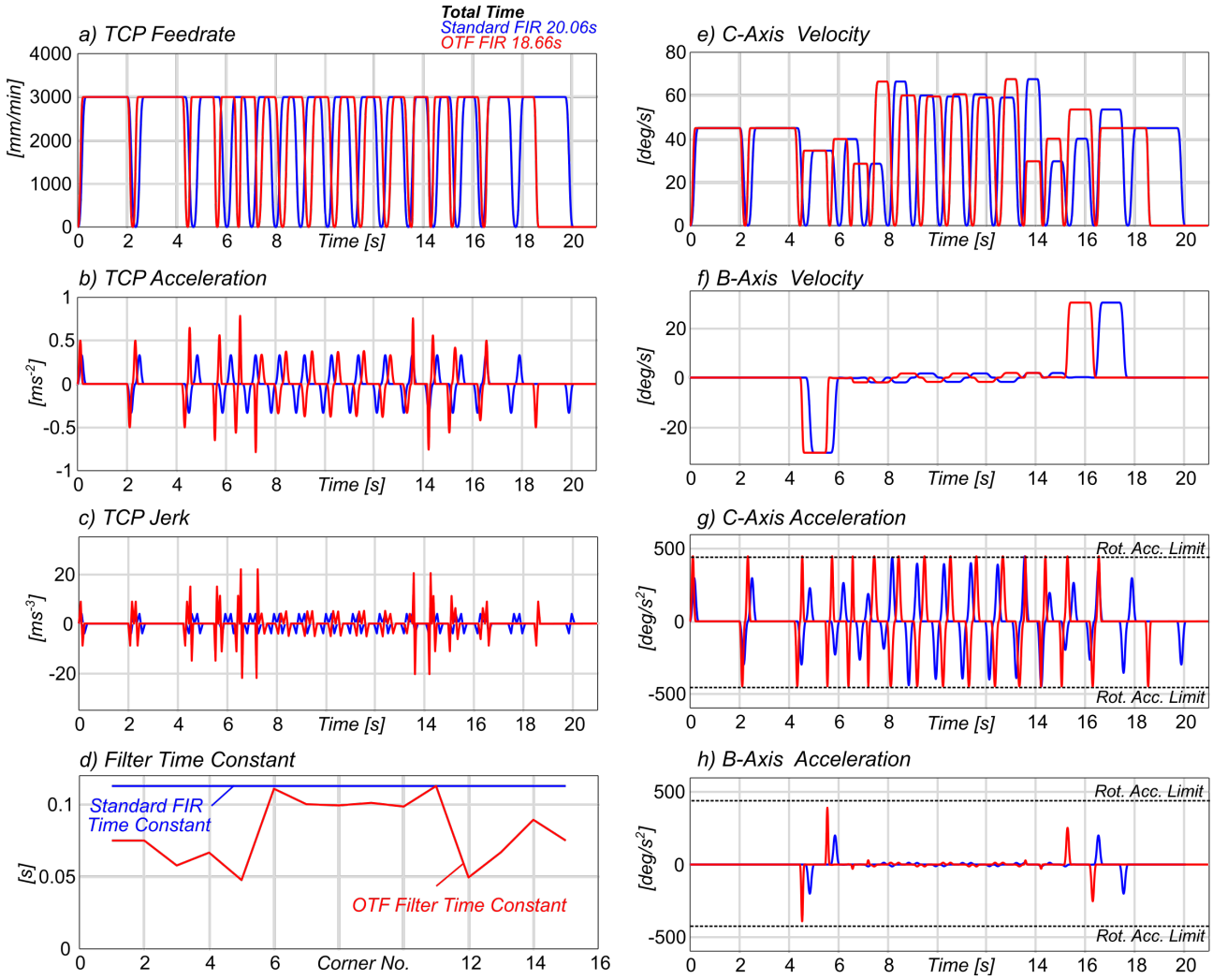

Figure 4.12: Kinematic profiles for P2P 5-axis motion generated from the proposed OTF interpolation method and compared with the standard FIR interpolation method

4.4.2 Non-stop Continuous 5-Axis Motion

The following section will demonstrate the case for continuous non-stop machining with user defined TCP and ORI constraints. For the illustrative example, the TCP and ORI tolerance was set at 50μm and 1° respectively. The results are presented in Fig.4.13.

Inspecting Fig.4.13a, the results show cycle times of 18.08s and 17.10s for the standard and OTF interpolation methods respectively, demonstrating a 5.42% reduction in machining cycle time for the OTF case.

Figure 4.13d shows the FIR time constant for the two cases. The FIR time constant for the standard FIR case does not change as this is traditionally selected based on the worse case feedrate and maximum acceleration or jerk. The figure shows the adaptive nature of the OTF FIR filter time constant in which for all but 2 of the CL-segments the kinematic response is faster than the standard FIR case. The benefit of this can be seen in Fig.4.13h. The OTF FIR method maximises the rotary acceleration of the C-axis to the limit for each rotary move, this is in contract to the standard FIR method which only reaches the limit once during the operation. The C-axis drive is always the limiting axes during this toolpath as the A-axis always remains within kinematic limits (Fig.4.13i).

Inspecting the TCP accuracy of the interpolation methods, Fig.4.13e shows the TCP blending errors are within limits and only very minor advantages are witnessed in the OTF FIR method. Similarly, Fig.4.13f shows the ORI blending errors well respect the 1° user defined limit. The reason for the significant over accuracy is due to the TCP tolerance taking precedence when selecting the overlap time T_o using (4.51).

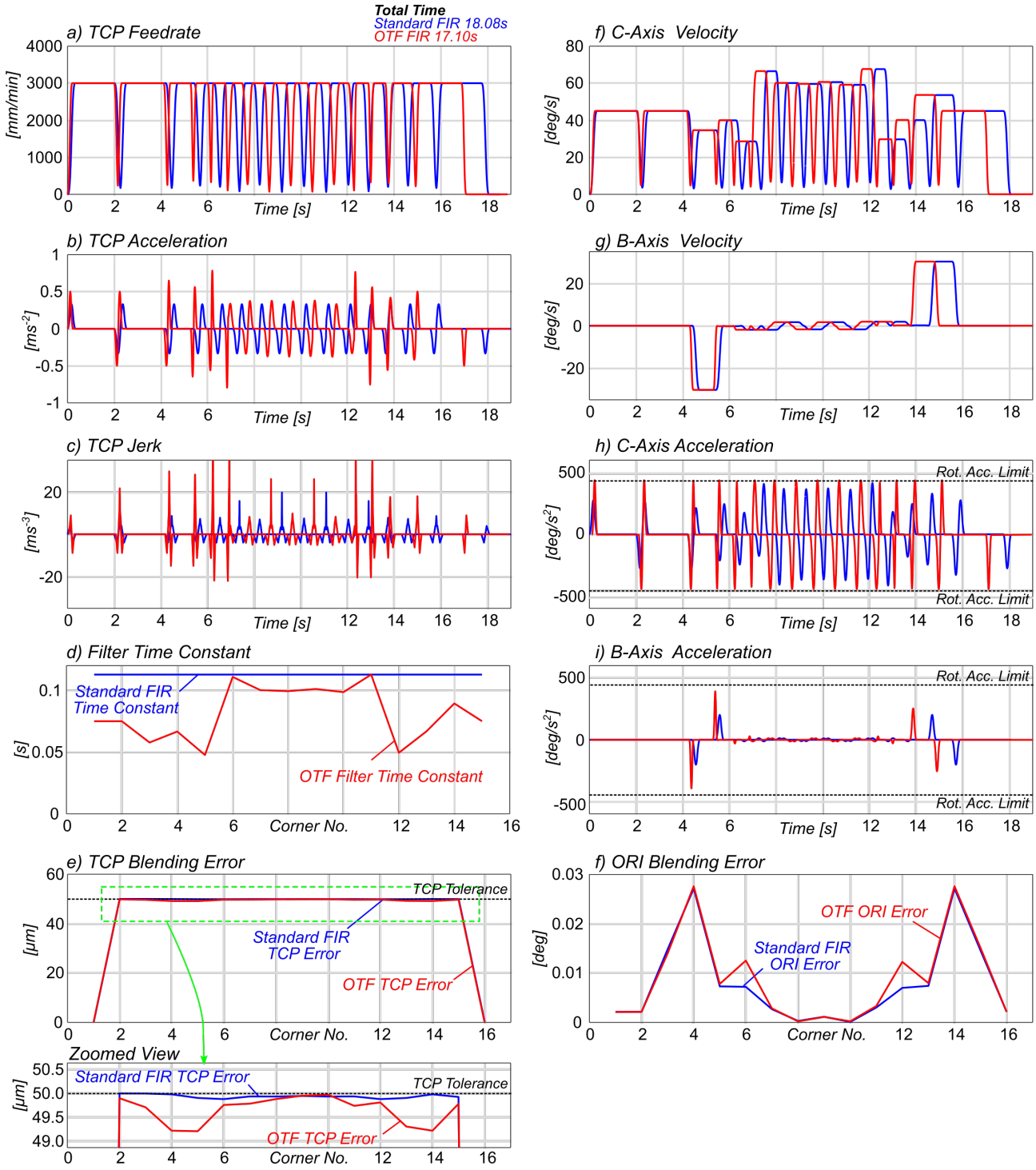

Figure 4.13: Kinematic profiles for continuous non-stop 5-axis motion generated from the proposed OTF interpolation method and compared with the standard FIR interpolation method

4.5 Conclusions

This paper presented a novel real-time on-the-fly interpolation technique for 3 and 5-axis machine tools. It introduced the use of an adaptive interpolation scheme based on finite impulse response filtering. The method introduced on-the-fly FIR filter time constants which through interpolation kinematically optimise each commanded segment during toolpath motion. A >5% and 7% reduction in machining cycle times was demonstrated when compared to standard FIR interpolation in 5-axis continuous and P2P machining respectively. The proposed method maximises feed drive performance for each CL command whilst satisfying TCP and ORI user defined tolerances. In 3-axis, approximately 4% and 7% reduction in cycle times was demonstrated during continuous and P2P machining respectively. Finally, The TCP and ORI blending errors were analytically modelled and confined using the overlap time of the reconstructed interpolated signals. For the first time, the consecutive feedrates were assumed different and analytically modelled resulting in accurate TCP and ORI errors and further reductions in cycle time when compared to previous conservative methods.

The main challenge during FIR based interpolation for non-stop continuous machining is controlling the TCP and axis jerk. Changing the structure of the velocity pulses is one method to achieve this as demonstrated in chapters 2 to 4. Further work should further investigate methods to optimise the jerk analytically to ensure fast NC interpolation.

This research demonstrated the ability to change kinematic responses during reference signal generation. This on-the-fly method opens up further opportunities for control of position or pose dependent dynamics in CNC and robotic machining.

Appendix 4A - Piecewise Analytical Kinematic Equations for P2P Motion

$$s'_{TCP}(t) = \begin{cases} \frac{F}{24T_{TCP}^3}t^4, & 0 \le t < T_{TCP} \\ -\frac{F}{24\,{T_{TCP}}^3}\left(3\,{T_{TCP}}^4 - 12\,{T_{TCP}}^3\,t + 18\,{T_{TCP}}^2\,t^2 - 12\,T_{TCP}\,t^3 + 2\,t^4\right), & T_{TCP} \le t < 2T_{TCP} \\ \frac{F}{24\,{T_{TCP}}^3}\left(45\,{T_{TCP}}^4 - 84\,{T_{TCP}}^3\,t + 54\,{T_{TCP}}^2\,t^2 - 12\,T_{TCP}\,t^3 + t^4\right), & 2T_{TCP} \le t < 3T_{TCP} \\ -\frac{F}{2}\left(3\,T_{TCP} - 2\,t\right), & 3T_{TCP} \le t < T_v \end{cases} \tag{4.52}$$

$$v'(t) = \begin{cases} \frac{F}{6T_{TCP}^3}t^3 & 0 \le t < T_{TCP} \\ \frac{F}{2T_{TCP}^3}\left(-\frac{2}{3}t^3 + 3T_{TCP}t^2 - 3T_{TCP}^2t + T_{TCP}^3\right) & T_{TCP} \le t < 2T_{TCP} \\ \frac{F}{2T_{TCP}^3}\left(\frac{1}{3}t^3 - 3T_{TCP}t^2 + 9T_{TCP}^2t - 7T_{TCP}^3\right) & 2T_{TCP} \le t < 3T_{TCP} \\ F & 3T_{TCP} \le t < T_v \end{cases} \tag{4.53}$$

$$a'(t) = \begin{cases} \frac{F}{2T_{TCP}^3}t^2 & 0 \le t < T_{TCP} \\ \frac{F}{T_{TCP}^3}\left(-t^2 + 3T_{TCP}t - \frac{3}{2}T_{TCP}^2\right) & T_{TCP} \le t < 2T_{TCP} \\ \frac{F}{T_{TCP}^3}\left(\frac{1}{2}t^2 - 3T_{TCP}t + \frac{9}{2}T_{TCP}^2\right) & 2T_{TCP} \le t < 3T_{TCP} \\ 0 & 3T_{TCP} \le t < T_v \end{cases} \tag{4.54}$$

$$j'(t) = \begin{cases} \frac{F}{T_{TCP}^3}t & 0 \le t < T_{TCP} \\ \frac{2F}{T_{TCP}^3}\left(-t + \frac{3}{2}T_{TCP}\right) & T_{TCP} \le t < 2T_{TCP} \\ \frac{F}{T_{TCP}^3}\left(t - 3T_{TCP}\right) & 2T_{TCP} \le t < 3T_{TCP} \\ 0 & 3T_{TCP} \le t < T_v \end{cases} \tag{4.55}$$

Appendix 4B - Star Shaped Toolpath CL Commands

Px	Py	Pz	Oi	Oj	Ok
0	0	0	0.5025	0.5025	0.7035
100	0	0	-0.5025	0.5025	0.7035
100	100	0	-0.5025	-0.5025	0.7035
50	100	10	0	-0.3511	0.9363
40	70	20	0.1562	-0.3123	0.9370
10	70	10	0.2357	-0.2357	0.9428
30	45	20	0.3162	0	0.9487
20	10	10	0.2357	0.2357	0.9428
50	30	20	0	0.3162	0.9487
80	10	10	-0.2357	0.2357	0.9428
70	45	20	-0.3162	0	0.9487
90	70	10	-0.2357	-0.2357	0.9428
60	70	20	-0.1562	-0.3123	0.9370
50	100	10	0	-0.3511	0.9363
0	100	0	0.5025	-0.5025	0.7035
0	0	0	0.5025	0.5025	0.7035

Table 4.1: 5-Axis star shaped toolpath TCP position and orientation commands

Chapter 5

Conference Paper - On-The-Fly CNC Interpolation using Frequency-Domain FFT-based Filtering

5.1 Introduction

Interpolation of CNC toolpaths using Finite Impulse Response (FIR) filtering has been proven a computationally efficient method of trajectory generation. Existing FIR based methods use direct convolution in the discrete domain to filter reference commands and generate smooth feed drive reference trajectories. Within these methods, the FIR filter time constant, which determines the kinematic properties of the reference signal, is calculated from the maximum commanded feedrate set by the part program. Therefore, for toolpaths with varying feedrate commands (such as feedrate scheduled toolpaths), the FIR filter is sub-optimal for all but the maximum feedrate commands leading to slower drive responses and longer overall machining cycle times. One method to overcome this limitation is to use On-The-Fly (OTF) interpolation and change the kinematic properties of the reference signal online. OTF interpolation optimises the reference signal for each change in feedrate along the toolpath and reduces the overall machining cycle time. Optimising the kinematics throughout the toolpath requires changing the FIR filter time constant online. However, this imposes significant computational challenges when filtering the long input signal. To date, all FIR-based filtering methods use the same FIR filter time constants throughout the toolpath and interpolate using direct convolution of a single input signal with the FIR filter [21, 49, 50, 62]. Using methods from real-time audio processing [63] we address this challenge.

By segmenting each cutter location (CL)-line of the G-code, the input signals can be split and filtered separately by an optimised FIR filter followed by signal reconstruction to form the full smoothed reference signals. The segmented filtering can be achieved either by direct convolution or via frequency domain methods. The convolution theorem states that multiplication in the frequency domain is equivalent to convolution in the time domain [64], and as such the use of Fast Fourier Transforms (FFTs) can offer greater performance computationally for real-time Numerical Control (NC) interpolation. The advantages of fast FFT convolution are seen when the original signals are long and subsequently segmented into smaller blocks. This is the case for circular convolution using the block transform method. The segmented filtered output signals are concatenated with an overlapped section where the values are added, this is termed Overlap-Add (OLA) signal reconstruction [64][63]. The result is the linear interpolation of the original input signal.

Unlike previous research [48][25], no single continuous pulse train signals are generated and subsequently filtered. The CL-blocks are individually filtered using frequency domain methods and converted back into the time domain. The OLA method of signal reconstruction is used to control the cornering feedrates which in turn are used to control the tool centre point (TCP) blending tolerances. The FIR filter is designed separately for each individual CL command and this provides on-the-fly filtering to optimise acceleration and jerk, resulting in a reduced machining cycle time compared to standard FIR based filtering. In this paper, we propose FFT based convolution as a computationally efficient method interpolating NC toolpaths whilst simultaneously handling kinematic and tolerance constraints.

5.2 Application to CNC Interpolation

This research implements the circular convolution (block transform) method to linearly interpolate each individual velocity pulse separately and reconstruct the final input signal from the convolved parts. The block transform method applied to CNC interpolation is presented in the following sections for point-to-point (P2P) and non-stop continuous TCP motion.

A P2P linear G01 move commands the tool motion from the starting TCP position $\mathbf{P}_s = [P_{s,x}, P_{s,y}, P_{s,z}]^T$ to the final TCP position defined by $\mathbf{P}_e$. In 3-axis machining the machining feedrate F is defined as the linear speed of the TCP. For each G01 command denoted by

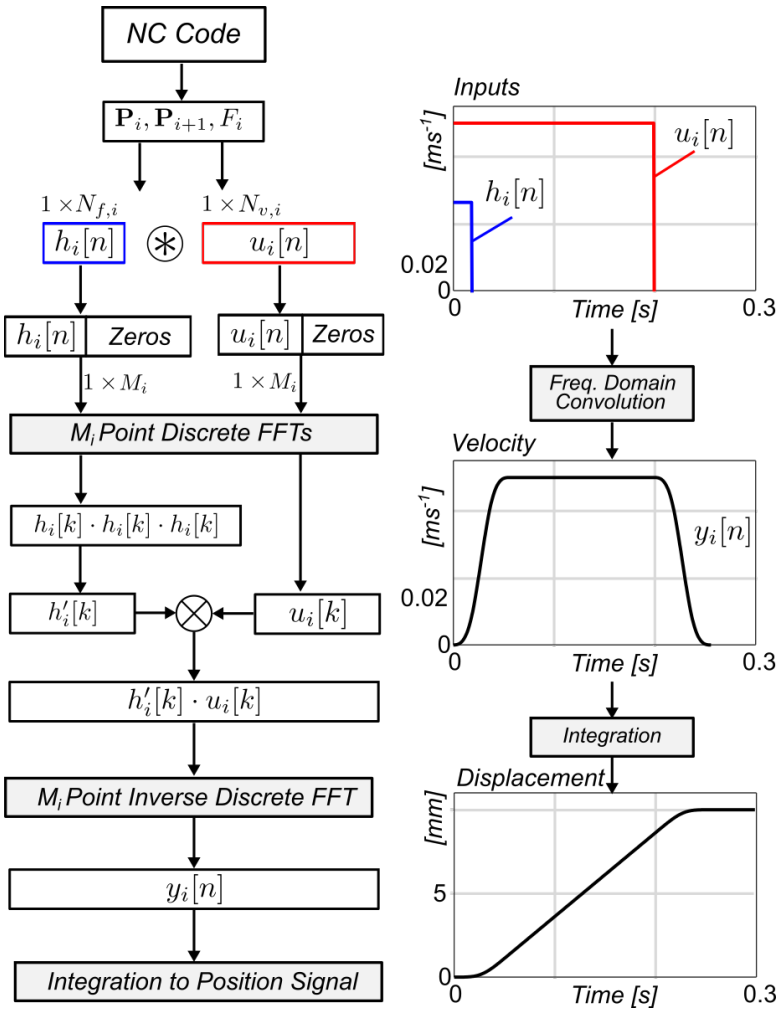

Figure 5.1: FFT-based interpolation method applied to a single CL-line

subscript i, where $i = [1, ..., N_{G01}]$ and N_{G01} is the number of G01 commands, the segmented input signals $u_i[n] \in \mathbb{R}^{1\text{x}N_{v,i}}$ are represented by a vector of discrete feedrate values F_i:

$$u_i[n] = [F_i, F_i, F_i, ...]^{1\text{x}N_{v,i}}, \tag{5.1}$$

where $N_{v,i} = T_{v,i}/T_s$, $T_{v,i} = L_i/F_i$ and $L_i = \|\mathbf{P}_{e,i} - \mathbf{P}_{s,i}\|_2$.
The FIR filter signal $h_i[n] \in \mathbb{R}^{1\text{x}N_{f,i}}$ is constructed as:

$$h_i[n] = \frac{1}{N_{f,i}} [[1, 1, 1, ...]^{1\text{x}N_{f,i}}, \tag{5.2}$$

where $N_{f,i} = T_{TCP,i}/T_s$. It is therefore $h_i(n)$ which controls the kinematic properties of the final interpolated signal based on $T_{TCP,i}$. Which, unlike in previous research where a single T_{TCP} was applied globally to the whole convolution operation based on a worst case feedrate, $T_{TCP,k}$ is now selected independently for each individual G01 command based on the segment feedrate F_i. Therefore each individual G01 response is optimised to ensure the maximum kinematic

performance. This allows "On-The-Fly" (OTF) filtering to change the machine tool kinematic response throughout the toolpath in real-time.

Next, in order to conduct FFT operations the segmented input signals $u_i(n)$ and FIR filter response vectors $h_i(n)$ are each padded with zeros to length M_i to prevent aliasing [65] when using overlap-add convolution. As a minimum, $M_i = N_{v,i}+3N_{f,i}-1$, however for computational efficiency, the length M_i should equal a power of 2, ($M_i = 2^x$, where $x = \text{ceiling}(\log_2(N_{v,i} + 3N_{f,i}))$ and account for the FIR filter to be applied 3 times to the input signal to generate a smooth C^3 continuous velocity signal.

$$u_{i,zp}[n] = \left[[F_i, F_i, F_i, ...]^{1\text{x}N_{v,i}}, [0, 0, 0, ...,]^{1\text{x}M_i - N_{v,i} - 1}\right], \tag{5.3}$$

$$h_{i,zp}[n] = \frac{1}{N_{f,i}}\left[[1, 1, 1, ...]^{1\text{x}N_{f,i}}, [0, 0, 0, ..]^{1\text{x}M_i - N_{f,i} - 1}\right], \tag{5.4}$$

where subscript zp denotes a zero padded vector.
The zero padded input signal $u_{i,zp}[n]$ and impulse response signal $h_{i,zp}[n]$ are transformed using M_i-point Discrete FFTs:

$$u_i[k] = FFT_{M_i}\left(u_{i,zp}[n]\right), \tag{5.5}$$

$$h_i[k] = FFT_{M_i}\left(h_{i,zp}[n]\right). \tag{5.6}$$

First, the impulse response $h_i[k]$ is multiplied 3 times in the frequency domain to achieve the equivalent 3 filter convolution:

$$h_i'[k] = h_i[k] \cdot h_i[k] \cdot h_i[k], \tag{5.7}$$

where $h_i'[k]$ represents the triple FIR filter. Next $h_i'[k]$ and $u_i[k]$ are multiplied in the frequency domain to achieve convolution:

$$y_i[k] = h_i'[k] \cdot u_i[k]. \tag{5.8}$$

Finally, an M_i-point inverse FFT is applied to $y_i[k]$ to generate the convolved segmented signal in the discrete time domain:

$$y_i[n] = IFFT_{M_i}\left(y_i[k]\right), \tag{5.9}$$

where $y_i[n] \in \mathbb{R}^{1\text{x}M_i}$. The resulting output signal $y_i(n)$ has been extended to M_i when compared to the segmented input signal $u_i[n]$ of length $N_{v,i}$. Only the first $N_{v,i} + 3N_{f,i}$ values of

the segmented signals are utilised in the signal reconstruction, therefore the segmented output signals are truncated and the remaining values (all zeros) are discarded. Finally, smoothed segmented velocity signal (5.9) is integrated to generate the interpolated position command. For P2P motion the individual position signals are concatenated end to end to generate the final reconstructed output signal. The end to end reconstruction generates an instantaneous stop at the start and end of each CL-line resulting in P2P motion.

Continuous TCP motion is achieved when the individual output signals are concatenated with an overlap to yield the final reconstructed output signal. In traditional OLA methods of signal reconstruction the final $3N_{f,i} - 1$ values of the output signal $y_i[n]$ would be added to the first $3N_{f,i} - 1$ values of the next output signal $y_{i+1}[n]$, hence the term "overlap-add". The overlap can be used as a method of constraining TCP contouring errors by controlling the cornering feedrates between two successive CL-lines (refer to chapter 4). The overlap is analytically determined based on consecutive feedrates, FIR filter time constants (set to satisfy acceleration and jerk limits) and TCP blending tolerance (and tool orientation blending tolerance in 5-axis machining). The generalised reconstructed reference signals are denoted by:

$$y'(n) = y'_i(n) \overset{T_{o,i}}{\bigcap} y'_{i+1}(n) \overset{T_{o,i+1}}{\bigcap} y'_{i+2}(n) \ldots y'_{N_{G01}}(n), \tag{5.10}$$

where $\overset{T_{o,i}}{\bigcap}$ represents the overlap of $T_{o,i}$ between adjacent segments during signal reconstruction.

5.3 Illustrative Case Studies

This section presents two case studies to demonstrate the computational efficacy of the FFT based OTF interpolation method. Each of the 2 case study toolpaths were interpolated 10^3 times to determine the average computation speeds.

5.3.1 5-Axis Spiral Semi-Finishing Toolpath

The first case study is based on a highly discritised 5-axis spiral toolpath as shown in Fig.5.2. The toolpath is commanded at 2000mm/min and composed of 6919 G01 commands where the average CL-line is 1.28 mm in length. The toolpath is interpolated on-the-fly using 3 different methods. The first uses segmented direct convolution implemented through difference equations, the second uses FFT based convolution, and the final method once again uses FFT convolution but with optimised input lengths. The computations were conducted on a 64-bit Intel(R) Xenon(R) 3.00GHz CPU. Table 5.1 shows the results from case study 1 and Figure 5.3 shows the resulting average computation times throughout the toolpath for each CL.

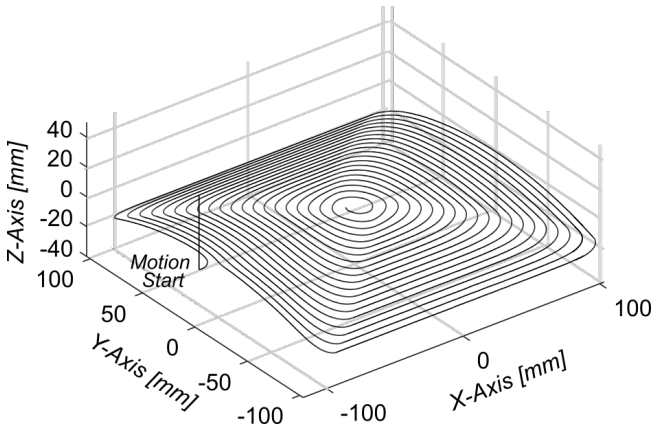

Figure 5.2: Illustrative Case Study: Spiral Toolpath TCP Trajectory

Inspecting Table 5.1, the mean computation time per output for the direct convolution method implemented via difference equations is 3.5×10^{-6} seconds. The two FFT methods are 3.7×10^{-7} and 3.4×10^{-8} seconds, corresponding to a 9.3 and 10.2 times improvement for the FFT and optimised FFT methods respectively.

Convolution Method	Mean Time Per Output (s)	Improvement (x times)
DC - Difference Equation	3.5×10^{-6}	
FFT, $M_i = N_{v,i} + 3N_{f,i} - 1$	3.7×10^{-7}	9.3
FFT, $M_i = 2^x$	3.4×10^{-7}	10.2

Table 5.1: Comparison of Computational Cost of Convolution Method for 5-Axis Finishing Spiral Toolpath

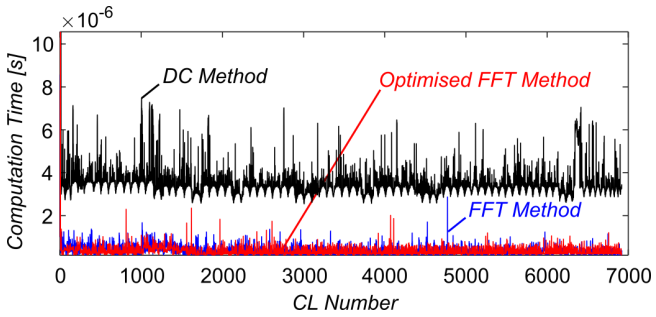

Figure 5.3: Spiral toolpath - computation times

5.3.2 3-Axis Roughing Toolpath

The second case study is based on a coursely discritised 3-axis roughing toolpath to show the impact of CL discritisation on the interpolation process. The roughing toolpath is shown in 5.4. The toolpath is commanded at 3000mm/min and is made up of 167 G01 commands averaging 112.3mm in length. From Table 5.2, the mean computation time per output for the direct convolution method implemented via difference equations is 5.0×10^{-6} seconds. The two FFT methods are 8.2×10^{-7} and 2.4×10^{-7} seconds, corresponding to a 6.1 and 10.6 times improvement for the FFT and optimised FFT methods respectively.

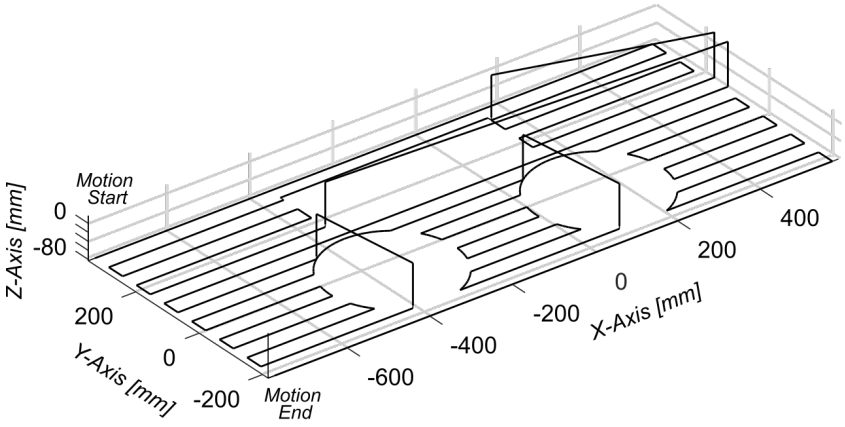

Figure 5.4: Illustrative Case Study: Roughing Toolpath TCP Trajectory

Table 5.2: Comparison of Computational Cost of Convolution Method for 3-Axis Roughing Toolpath

Convolution Method	Mean Time Per Output (s)	Improvement (x times)
DC - Difference Equation	5.0×10^{-6}	-
FFT, $M_i = N_{v,i} + 3N_{f,i} - 1$	8.2×10^{-7}	6.1
FFT, $M_i = 2^x$	2.4×10^{-7}	10.6.

5.4 Discussion

As demonstrated in section 5.3 there are significant advantages to using frequency domain methods for NC interpolation when compared to the difference equation implementation. Further advantages are gained when optimising the input lengths to base 2 integers. The computational cost of direct convolution using difference equations scales to $O\left(M_i^2\right)$ compared to the $O\left(M_i \log_2 M_i\right)$ for FFT computation. For $M_i > 64$ in length FFT based convolution offers significant advantages over direct convolution. As interpolator clock cycles become faster it is important to note that the lengths of input and filter signals will increase accordingly. Therefore implementing computational efficient methods of NC interpolation is vital.

5.5 Conclusions

Until recently, published research in FIR based NC interpolation has not considered adapting the FIR filter time constant throughout the toolpath. Newly introduced On-The-Fly methods can optimise the kinematic properties of the tool motion for the whole toolpath rather than based on a worst case feedrate scenario. This paper has proposed a novel method of OTF interpolation which also addresses the real-time computational cost of the algorithm. The main findings of the work are:

- On-The-Fly interpolation of NC toolpaths can be conducted by segmenting input signals and individually linearly interpolating each CL-line using convolution in the frequency domain.
- FFT based methods of convolution demonstrate significantly faster computational times per interpolated output than direct convolution methods.
- FFT based methods can be further optimised to reduce time of computation by selecting input and filter lengths to match exponential to the base 2 whole numbers.

Further work will investigate the frequency domain approach to locally control the frequency spectrum of the interpolated position reference commands.

Chapter 6

Conclusions

6.1 Key Findings

This thesis has presented an original and coherent body of work, which through systematic study has demonstrated an addition to knowledge in the field of CNC interpolation and motion control. The conclusions are hereby summarised.

6.1.1 Machining Feedrate and Cycle Time Prediction

Understanding the process chain from CAD/CAM to post processor, NC code, feed drive commands and finally TCP motion is important. A one-to-one mapping does not exist between what the process planner is requesting and the final machined product. This research has sought to understand this process and in particular it has focused on the link between NC code and feed drive commands, which has the greatest impact on the actual feedrate and overall machining cycle time.

To understand this gap, this research generated a new method of NC interpolation and trajectory generation which was applied to feedrate and cycle time prediction. A FIR filtering-based method of interpolation to overcome the computation burden of polynomial based methods was proposed. The design simplified the setup to a single design parameter for the kinematic constraints by using matched FIR filter time constants, thereby removing the requirement for a priori system testing. Methods for NC interpolation using both 2 and 3 FIR filters were developed and it was demonstrated using 3 matched FIR filters lowered machining cycle times compared to 2.

For the first time, a proposed offline feedrate and cycle time prediction method considered both TCP tolerance and drive kinematic limits. The method was validated against realistic industrial toolpaths such as pocketing and dynamic milling strategies including trochoidal toolpaths. The method was over 90% accurate and far exceeded current CAM-based predictions for machining cycle times. The proposed method applied to cycle time prediction in particular significantly benefits process and business planning.

The model was integrated into an advanced cutting force model to demonstrate the effectiveness of using accurate feedrate predictions when simulating cutting forces in complex toolpaths. The models were validated through experimental machining trials. Thereby, demonstrating the utility of the method for virtual machining and offline process optimisation.

6.1.2 MCS-WCS Interpolation

The natural progression was to extend the analytical and FIR-based interpolation methods from 3 to 5-axis machining. As the default setting on modern machining centres interpolates NC code in the MCS, the research focused on interpolating rotary axes in the MCS and TCP in the WCS. Thus, a joint workpiece-machine coordinate system interpolation method based on FIR filtering was introduced. In particular the method overcame the challenges associated with interpolating in spherical coordinates, such as rapid feed fluctuations when interpolating near kinematic singularities and complex real-time calculations.

The research addressed the problem of nonlinear interpolation errors caused by interpolation of the rotary position commands in the MCS. Using techniques from maritime navigation, the modified haversine formula was proposed to calculate nonlinear interpolation errors caused by spherical geometry. Using this technique, a method to discritise the ORI toolpath to satisfy user defined ORI tolerance was demonstrated and validated.

Building on the analytical work of the 3-axis interpolation method, the blending errors during non-stop continuous interpolation of linear 5-axis toolpaths were modelled and confined using the TCP and rotary feedrate scaling factor.

Finally, up to 10% reductions in machining cycle times were demonstrated by decoupling the Cartesian and rotary FIR filter time constants, ensuring both Cartesian and rotary feed drive

performance were maximised whilst satisfying TCP and ORI tolerances.

6.1.3 On-The-Fly Trajectory Generation

Previous methods of FIR-based interpolation used a single FIR filter time constant selected based on a worse case maximum feedrate and axis acceleration or jerk limit. This ensured the kinematic limits were not exceeded throughout the toolpath motion. However, this also resulted in sub-optimal cycle times as the FIR filter time constant was larger than necessary for all but the maximum feedrate commands. Addressing this issue, an adaptive interpolation scheme based on FIR filtering was proposed. The method introduced on-the-fly FIR filter time constants which through interpolation kinematically optimised each commanded segment during toolpath motion. Significant reductions in machining cycle times were demonstrated for both P2P and continuous non-stop machining when compared to the standard FIR case.

New analytical methods were used to constrain the TCP and ORI blending errors during FIR interpolation and for the first time, consecutive feedrates were assumed to be different. The analytical methods allow for fast computation and implementation without the requirement for iterative methods.

6.1.4 Frequency-Domain FFT-based Filtering

The benefits of the presented FIR-based methods of NC interpolation is the speed of computation when using analytical methods to satisfy machining constraints. Research to date has focused on using direct convolution to interpolate NC programs in the time domain. Based on the convolution theorem which states multiplication in the frequency domain is equivalent to convolution in the time domain, this research demonstrated interpolation of NC reference commands using FFTs is equally as effective as interpolation using direct convolution methods. The benefit of using frequency-domain convolution showed a significant increase in computation speed per interpolated output. The ability to use FFTs to interpolate and adaptively modify kinematic properties of generated reference trajectories online opens up further avenues of research in controlling position and pose dependent dynamics of CNC and robotic machine tools.

6.2 Industrial Practise Recommendations

Based on the research presented in this thesis, the following section highlights recommendations for industrial practice and implementation.

6.2.1 Chapter 2 Recommendations

1. The developed machining cycle time estimation software requires the part program, TCP tolerance and machine tool jerk constraint. The TCP tolerance is set by the CAM engineer/ process planner when creating the part program however the machine tool jerk value is not directly defined. The options available to the engineer/planner, depending on the controller, are of the form roughing, semi-finishing or finishing. These parameters set the maximum acceleration or jerk depending on the required surface finish. The associated value can be found from the machine tool technical manual or by contacting the OEM.

2. If using commercial CAM packages to estimate machining cycle times it is important to check if the TCP tolerance is accounted for. Most modern packages do not currently include this, therefore take caution with their estimates.

3. Similar principles apply to virtual machining packages when simulating cutting forces. The cutting force estimations will not be accurate if the models do not account for interpolator dynamics.

4. The effect of TCP tolerance on machining cycle times has been demonstrated in this chapter. It is recommended that engineers/planners are cognisant of the machining objective when setting machining tolerances. Sub-optimal tolerances could lead to significantly longer machining cycle times and reduce productivity.

6.2.2 Chapter 3 Recommendations

1. The effect of nonlinear interpolation errors during MCS interpolation has been demonstrated. Process planners and machine tool operators must be cognisant of the type of 5-axis interpolation method defined by the part program and also the machine tool default settings.

2. The proposed method of minimising the nonlinear interpolation errors could be used within CAM systems during the toolpath generation stage. Using within the CAM stage

of the production process would reduce tool orientation errors independent of the type of on-machine interpolation method.

3. Significant improvements in cycle times were demonstrated by decoupling the TCP and rotary FIR time constants. For machine tool builders, implementation of the decoupled approach is recommended for roughing operations in which minor deviations in TCP/ORI synchronisation are permissible.

6.2.3 Chapter 4 Recommendations

1. It has been demonstrated that using a static FIR filter time constant reduces the available performance of the feed drives. For machine tool builders, it is recommended that on-the-fly interpolation is implemented to reduce machining cycle times - in particular during roughing operations.

2. This research has focused on reducing machining cycle times by maximising drive performance through satisfying kinematic constraints. However, acceleration and jerk limits also have a direct impact on surface quality. It is recommended machine tool builders implement on-the-fly interpolation as a method of optimising surface condition.

6.2.4 Chapter 5 Recommendations

1. Using FFT based interpolation has been demonstrated computationally efficient. For machine tool builders, it is recommended to implement the on-the-fly FFT based interpolation methods. Benefits would include enabling a longer lookahead function and preventing any computational bottlenecks caused from overloading the buffer.

2. The computational efficiencies of the FFT-based method would enable machine tool controller manufacturers to increase interpolation frequencies. Increased frequencies would allow numerical controller functions to measure and control the higher order dynamics of the machine tool such as structural vibrations and chatter. In-process measurement and control functions are favourable to external sensor based systems as they reduce risk of damage, maintenance and costs.

6.3 Recommendations for Future Work

The research presented in this thesis has positively contributed to the development of FIR filtering based CNC interpolation techniques. Building upon the presented research, this section presents the recommendations for further work.

1. This thesis focused on analytical methods of determining tool motion to control TCP and ORI contouring errors during 3 and 5-axis local corners. Further work should be extended to tool motion behaviour during 3 and 5-axis global corners where significant overlap and kinematic coupling exists between motion generated from multiple highly discritised CL-lines.

2. The use of analytical representations of displacement and velocity to calculate FIR interpolation parameters is the bread and butter of this thesis. The method holds well for precision tolerances, however, for more relaxed tolerances such as for roughing operations the solutions to the analytical equations become more complex. Further work should extend the analytical methods to these machining conditions.

3. Two different 5-axis methods of NC interpolation were presented in this thesis and each contributed elements to reducing machining cycle times, namely decoupling the Cartesian and rotary FIR filter time constants in the chapter 3 paper, and introducing an adaptive OTF FIR filter time constants in chapters 4 and 5. The next step is to combine these methods to further improve cycle times and speed up the production process

4. As this research has demonstrated, On-The-Fly methods of CNC interpolation show great potential of tuning interpolation parameters online throughout the toolpath. In particular, further work should address optimising the interpolation by addressing the change in dynamic properties of the machine tool/workpiece online. For example one application is to the changing torque limit on the rotary axis acceleration limit in response to the changing mass of a heavy workpiece during roughing operations.

5. Finally, the research in this thesis focused on 5-axis CNC machine tools. However, the research translates across to many precision motion control applications. In particular, it is recommended that the methods in this thesis are extended to robotic platforms with the aim of further developing robotic machining capabilities.

Appendix A

DMU Evo40 Non-Orthogonal Kinematic Modelling

A.1 Kinematics of Non-Orthogonal 5 Axis Machining

The following section presents modelling work undertaken during the period of the EngD.

The forward and inverse kinematics for the DMU eVo 40 are more complex than the standard A/C trunnion table type machining centre. Much like for the standard configuration, the forward and inverse kinematic chains are modelled by a series of coordinate transformations which shift the axis systems from one axis system or coordinate frame to another. The key difference been there are more coordinate systems and shifts to consider. This appendix will demonstrate the method of calculating the forward and inverse kinematics for the non-orthogonal machining centre configurations.

The DMU eVo 40 is a 5-axis machining centre with a non-orthogonal axis configuration. As shown in Fig A.1, the B axis is negatively rotated about the machine Z-axis by 45. The pivot axis (B) and rotary axis (C) intersect at the machine pivot point, located $d = 125$mm above the base coordinate frame and there is a 55° angle between the B and C-axis.

The transformation are conducted using Homogeneous Transformation Matrices (HTMs) [66]. These are 4x4 matrices that allow coordinate system rotation and translation, as represented in (A.1).

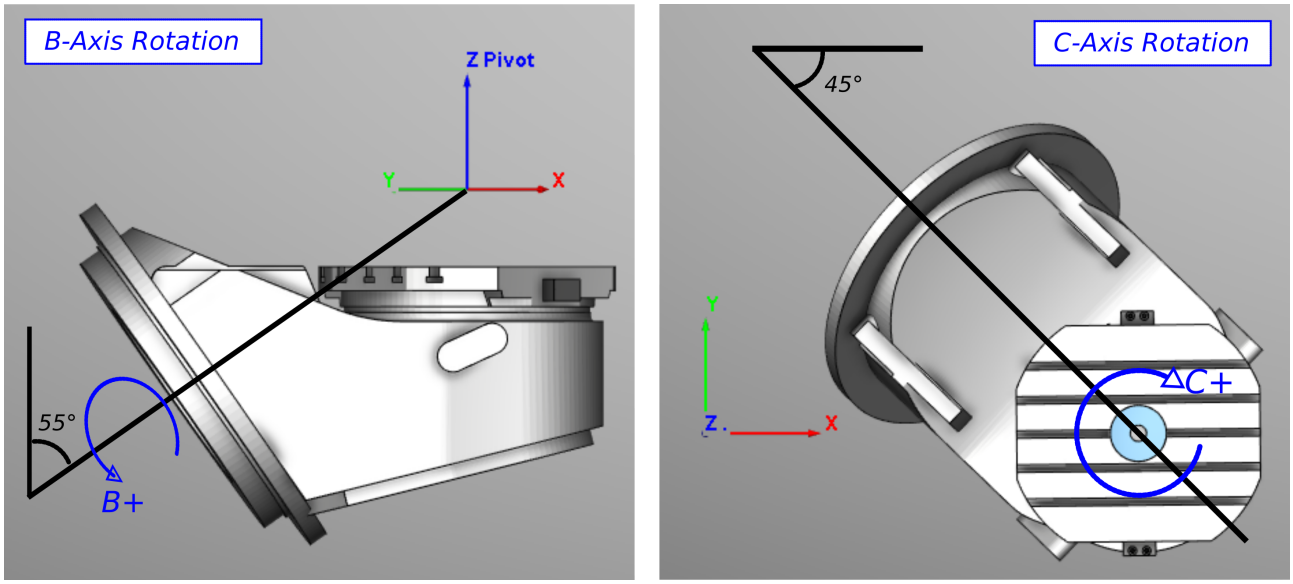

Figure A.1: B and C Axes Configurations - DMU eVo 40

$$H_n^p = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{A.1}$$

where a shift in coordinate system from p to n is represented by the HTM H_n^p. The first 3 columns denote the angular rotation in i,j,k parameters of the local x,y, and z axis respectively. The final column represents linear translations of the axis centre. The HTMs that constitute the kinematic model of the eVo 40 machining centre can be found in Appendix A.2.

The kinematic model of the eVo 40 requires 11 coordinate systems/frames and 10 coordinate transformations, these are are shown in Figs A.2 and A.3.

A.1.1 Forward Kinematics

The objective of the forward kinematics is to calculate WCS variables P_x, P_y, P_z,O_i, O_j, O_k from the MCS variables $X, Y, Z, \theta_B, \theta_C$. The coordinate transformation from MCS to WCS is denoted by the HTM H^{mcs}_{wcs}, and is calculated by the kinematic chain of HTMs as follows:

$$H^{mcs}_{wcs} = H^{9}_{wcs} H^{8}_{9} H^{7}_{8} H^{6}_{7} H^{5}_{6} H^{4}_{5} H^{3}_{4} H^{2}_{3} H^{1}_{2} H^{0}_{1} H^{mcs}_{0}, \tag{A.2}$$

and, $H^{wcs}_{mcs} = (H^{mcs}_{wcs})^{-1}$, is calculated from the inverse HTM transformations:

$$\begin{aligned} H^{mcs}_{wcs} \\ &= (H^{wcs}_{9})^{-1} \\ &\left(H^{9}_{8}\right)^{-1} \\ &\left(H^{8}_{7}\right)^{-1} \\ &\left(H^{7}_{6}\right)^{-1} \\ &\left(H^{6}_{5}\right)^{-1} \\ &\left(H^{5}_{4}\right)^{-1} \\ &\left(H^{4}_{3}\right)^{-1} \\ &\left(H^{3}_{2}\right)^{-1} \\ &\left(H^{2}_{1}\right)^{-1} \\ &\left(H^{1}_{0}\right)^{-1} H^{mcs}_{0}. \end{aligned} \tag{A.3}$$

The HTM H^{mcs}_{wcs} represents the tool position and orientation in the WCS. As shown in (A.1), the TCP position and Z-axis orientation are given in the fourth and third columns respectively. Defining the z-axis orientation in the MCS as $O_k = 1$, then the WCS parameters can be calculated using the forward kinematic chain, (A.3), as follows:

$$\begin{bmatrix} O_i & P_x \\ O_j & P_y \\ O_k & P_z \\ 0 & 1 \end{bmatrix} = H^{mcs}_{wcs} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{A.4}$$

Using (A.4), the analytical solutions to the O_i,O_j and O_k tool WCS orientations are presented as:

$$\begin{aligned} O_i = \sin(55)(&\sin(45)\sin\theta_B\sin\theta_C - \cos(45)\cos(55)\sin\theta_C - \cos(55)\cos\theta_C\cos(45) \\ &- \cos(45)\cos\theta_C\sin\theta_B\cos(45)\cos(55)\cos\theta_B\sin\theta_C + \cos(55)\cos\theta_B\cos\theta_C\sin(45)), \end{aligned} \tag{A.5}$$

$$\begin{aligned} O_j = -\sin(55)\,(&\cos(55)\,\sin(45)\,\sin\theta_C + \cos(45)\,\sin\theta_B\,\sin\theta_C + \cos\theta_C\,\sin(45)\,\sin\theta_B \\ &- \cos(45)\,\cos(55)\,\cos\theta_C + \cos(45)\,\cos(55)\,\cos\theta_B\,\cos\theta_C - \cos(55)\,\cos\theta_B\,\sin(45)\,\sin\theta_C), \end{aligned} \tag{A.6}$$

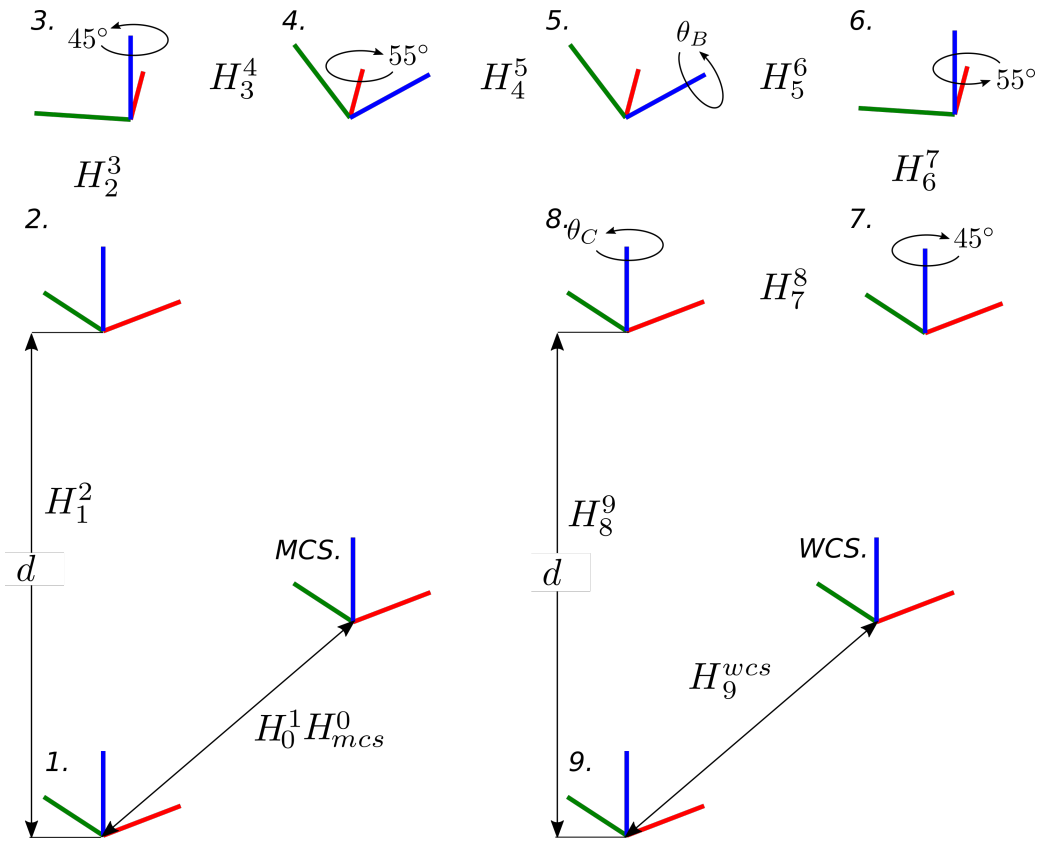

Figure A.2: Coordinate systems for modeling non-orthogonal 5-axis machine tool kinematics

$$O_k = \cos\theta_B + \cos^2(55) - \cos\theta_B \cos^2(55). \tag{A.7}$$

Due to the size of the equations, the analytical solutions to P_x,P_y and P_z TCP positions for the non-orthogonal 5-axis configuration are presented in appendix A.2 as equations (A.33),(A.34) and (A.35) respectively.

The relationship between the rotary axis velocities $\dot{\theta}_B$,$\dot{\theta}_C$, and the angular velocities $\dot{O}_i$,$\dot{O}_j$,$\dot{O}_k$ is defined by the orientation Jacobian as

$$\begin{bmatrix} \dot{O}_i \\ \dot{O}_j \\ \dot{O}_k \end{bmatrix} = \begin{bmatrix} \frac{\partial O_i}{\partial \theta_B} & \frac{\partial O_i}{\partial \theta_C} \\ \frac{\partial O_j}{\partial \theta_B} & \frac{\partial O_j}{\partial \theta_C} \\ \frac{\partial O_k}{\partial \theta_B} & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_B \\ \dot{\theta}_C \end{bmatrix}. \tag{A.8}$$

A.1.2 Inverse Kinematics

The inverse kinematics are used to calculate the MCS commands from the CL data in the part program (WCS).

$$(H_{wcs}^{mcs})^{-1} = H_{mcs}^{wcs}. \tag{A.9}$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix} = H_{mcs}^{wcs} \begin{bmatrix} O_i & P_x \\ O_j & P_y \\ O_k & P_z \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{A.10}$$

The B-axis inverse solution can be found from (A.7):

$$\theta_B = \arccos\left(\frac{k - cos^2(55)}{1 - cos^2(55)}\right). \tag{A.11}$$

The C-Axis is calculated using (A.10) where $a_{31} = k = 1$.

$$\begin{aligned} a_{31} &= 1 \\ &= O_k \left(\cos(55)^2 + \cos\theta_B \sin(55)^2\right) \\ &\quad - O_j \left(\cos(55)\sin(45)\sin(55)\sin\theta_C - \cos(45)\cos(55)\cos\theta_C\sin(55)\right. \\ &\qquad + \cos(45)\sin(55)\sin\theta_B\sin\theta_C + \cos\theta_C\sin(45)\sin(55)\sin\theta_B \\ &\qquad \left. + \cos(45)\cos(55)\cos\theta_B\cos\theta_C\sin(55) - \cos(55)\cos\theta_B\sin(45)\sin(55)\sin\theta_C\right) \\ &\quad - O_i \left(\cos(45)\cos(55)\sin(55)\sin\theta_C + \cos(55)\cos\theta_C\sin(45)\sin(55)\right. \\ &\qquad + \cos(45)\cos\theta_C\sin(55)\sin\theta_B - \sin(45)\sin(55)\sin\theta_B\sin\theta_C \\ &\qquad \left. - \cos(45)\cos(55)\cos\theta_B\sin(55)\sin\theta_C - \cos(55)\cos\theta_B\cos\theta_C\sin(45)\sin(55)\right). \end{aligned} \tag{A.12}$$

Using the trigonometric identity $\cos^2\theta + \sin^2\theta = 1$, equation (A.12) can be written in the form,

$$a'\sin\theta_C + b'\cos\theta_C = 1, \tag{A.13}$$

where $a' = a/d$, $b' = b/d$,

$$\begin{aligned} a &= -\sin(55)\left(O_i\cos(45)\cos(55) + O_j\cos(55)\sin(45) + O_j\cos(45)\sin\theta_B - O_i\sin(45)\sin\theta_B\right. \\ &\qquad \left. - O_i\cos(45)\cos(55)\cos\theta_B - O_j\cos(55)\cos\theta_B\sin(45)\right), \end{aligned} \tag{A.14}$$

$$\begin{aligned} b &= O_j\cos(45)\cos(55)\sin(55) - O_i\cos(55)\sin(45)\sin(55) \\ &\quad - O_i\cos(45)\sin(55)\sin\theta_B - O_j\sin(45)\sin(55)\sin\theta_B \\ &\quad - O_j\cos(45)\cos(55)\cos\theta_B\sin(55) + O_i\cos(55)\cos\theta_B\sin(45)\sin(55), \end{aligned} \tag{A.15}$$

and,

$$d = 1 - O_k \left(\cos^2(55) + \sin^2(55)\cos\theta_B\right). \tag{A.16}$$

Using the four-quadrant inverse tangent, the solution to θ_C for $-180° < \theta'_C \leq 180°$, is calculated by

$$\theta'_C = \arctan(a', b') = \arctan(a, b). \tag{A.17}$$

The C-axis is commanded range is $0° < \theta'_C \leq 360°$, therefore,

$$\theta_C = \theta'_C + 180°. \tag{A.18}$$

The expression for X,Y and Z are calculated from (A.10), where $a_{12} = a_{22} = a_{32} = 0$. The equations are of the form:

$$X = \alpha_1 P_x + \alpha_2 P_y + \alpha_3 P_z + \alpha_4 G_{54,X} + \alpha_5 x_{\text{Offset}} + \alpha_6 y_{\text{Offset}} + \alpha_7 z_{\text{Offset}} + \textit{trig terms}, \tag{A.19}$$

$$Y = \alpha_1 P_x + \alpha_2 P_y + \alpha_3 P_z + \alpha_4 G_{54,Y} + \alpha_5 x_{\text{Offset}} + \alpha_6 y_{\text{Offset}} + \alpha_7 z_{\text{Offset}} + \textit{trig terms}, \tag{A.20}$$

$$Z = d + \alpha_1 P_x + \alpha_2 P_y + \alpha_3 P_z + \alpha_4 G_{54,Z} + \alpha_5 x_{\text{Offset}} + \alpha_6 y_{\text{Offset}} + \alpha_7 z_{\text{Offset}} + \textit{trig terms}. \tag{A.21}$$

The full equations (A.36),(A.37) and (A.38) are included in appendix A.4.

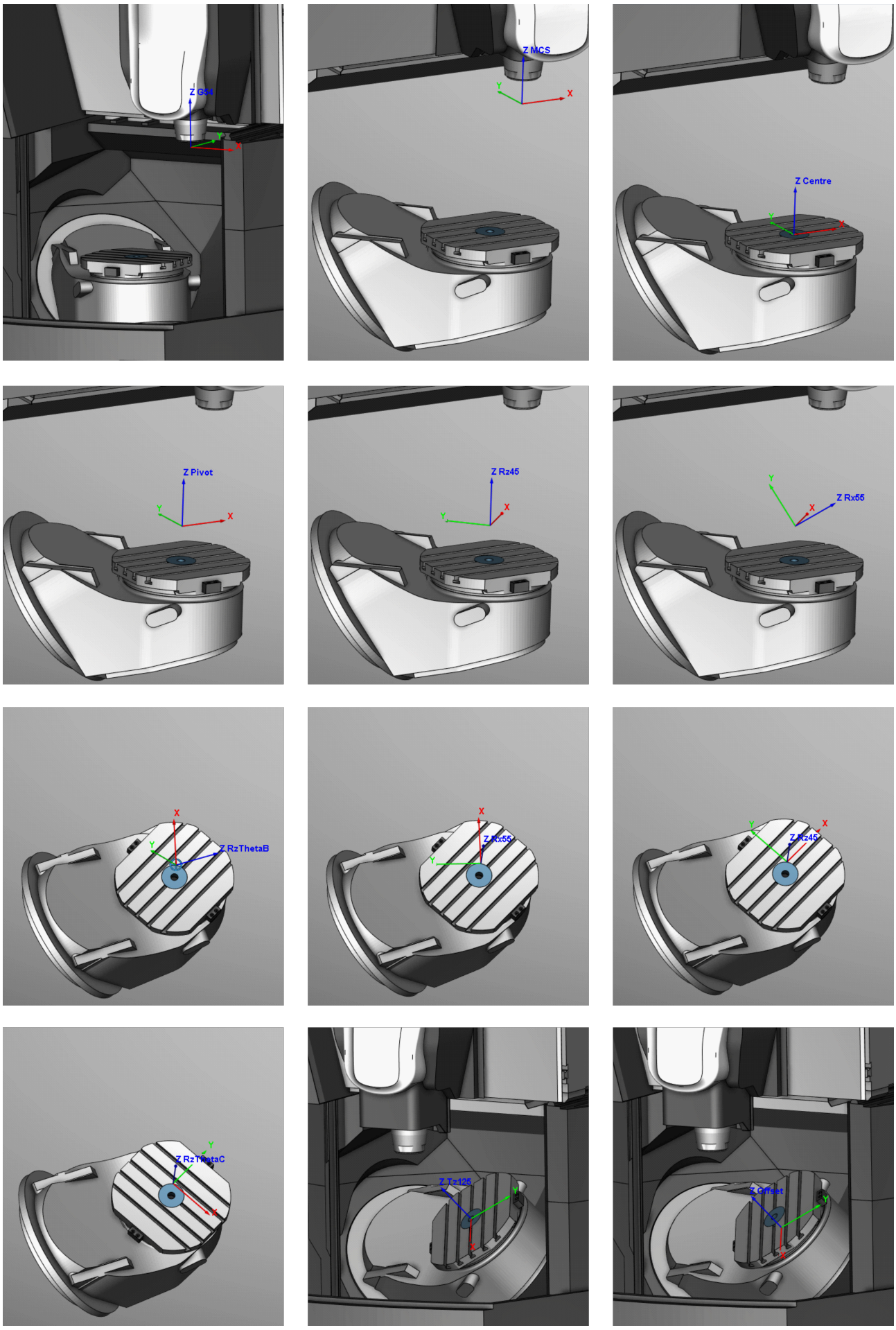

Figure A.3: Axis Systems for DMU eVo 40 Machining Centre with Non-Orthogonal Axis Configuration

A.2 DMU eVo 40 Homogeneous Transformation Matrices

$$H_0^{mcs} = \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.22}$$

$$H_0^1 = \begin{bmatrix} 1 & 0 & 0 & X_{G54} \\ 0 & 1 & 0 & Y_{G54} \\ 0 & 0 & 1 & Z_{G54} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.23}$$

$$H_1^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.24}$$

$$H_2^3 = \begin{bmatrix} \cos(45°) & -\sin(45°) & 0 & 0 \\ \sin(45°) & \cos(45°) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.25}$$

$$H_3^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(55°) & \sin(55°) & 0 \\ 0 & -\sin(55°) & \cos(55°) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.26}$$

$$H_4^5 = \begin{bmatrix} \cos(\theta_B^\circ) & -\sin(\theta_B^\circ) & 0 & 0 \\ \sin(\theta_B^\circ) & \cos(\theta_B^\circ) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.27}$$

$$H_5^6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-55°) & \sin(-55°) & 0 \\ 0 & -\sin(-55°) & \cos(-55°) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.28}$$

$$H_6^7 = \begin{bmatrix} \cos(-45^\circ) & \sin(-45^\circ) & 0 & 0 \\ \sin(-45^\circ) & \cos(-45^\circ) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.29}$$

$$H_7^8 = \begin{bmatrix} \cos(-\theta_C) & -\sin(-\theta_C) & 0 & 0 \\ \sin(-\theta_C) & \cos(-\theta_C) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.30}$$

$$H_8^9 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.31}$$

$$H_9^{wcs} = \begin{bmatrix} 1 & 0 & 0 & x_{\text{Offset}} \\ 0 & 1 & 0 & y_{\text{Offset}} \\ 0 & 0 & 1 & z_{\text{Offset}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.32}$$

A.3 Tool Position - Workpiece Coordinate System

$$
\begin{aligned}
x = {} & G_{54,Y}\cos(45)^2\,\sin\theta_C - G_{54,X}\,\cos\theta_C\sin(45)^2 - x_{Offset} + X\,\cos\theta_C\sin(45)^2 - Y\cos(45)^2\,\sin\theta_C \\
& - X\cos(55)^2\,\cos\theta_C\sin(45)^2 + Y\cos(45)^2\cos(55)^2\,\sin\theta_C + G_{54,Y}\,\cos(45)\,\cos\theta_C\,\sin(45) \\
& - G_{54,X}\,\cos(45)\,\sin(45)\,\sin\theta_C - Y\,\cos(45)\,\cos\theta_C\,\sin(45) + X\,\cos(45)\,\sin(45)\,\sin\theta_C \\
& - G_{54,X}\cos(45)^2\,\cos\theta_B\,\cos\theta_C + X\cos(45)^2\,\cos\theta_B\,\cos\theta_C + G_{54,Y}\,\cos\theta_B\sin(45)^2\,\sin\theta_C \\
& - Y\,\cos\theta_B\sin(45)^2\,\sin\theta_C + G_{54,X}\cos(55)^2\,\cos\theta_C\sin(45)^2 - G_{54,Y}\cos(45)^2\cos(55)^2\,\sin\theta_C \\
& - G_{54,Y}\,\cos(45)\,\cos\theta_B\,\cos\theta_C\,\sin(45) + G_{54,X}\,\cos(45)\,\cos\theta_B\,\sin(45)\,\sin\theta_C \\
& + G_{54,Z}\,\cos(45)\,\cos(55)\,\sin(55)\,\sin\theta_C + G_{54,Z}\,\cos(55)\,\cos\theta_C\,\sin(45)\,\sin(55) \\
& + G_{54,Z}\,\cos(45)\,\cos\theta_C\,\sin(55)\,\sin\theta_B + Y\,\cos(45)\,\cos\theta_B\,\cos\theta_C\,\sin(45) \\
& - X\,\cos(45)\,\cos\theta_B\,\sin(45)\,\sin\theta_C - Z\,\cos(45)\,\cos(55)\,\sin(55)\,\sin\theta_C \\
& - Z\,\cos(55)\,\cos\theta_C\,\sin(45)\,\sin(55) - Z\,\cos(45)\,\cos\theta_C\,\sin(55)\,\sin\theta_B \\
& + d\,\cos(45)\,\cos(55)\,\sin(55)\,\sin\theta_C + d\,\cos(55)\,\cos\theta_C\,\sin(45)\,\sin(55) \\
& + d\,\cos(45)\,\cos\theta_C\,\sin(55)\,\sin\theta_B - G_{54,Z}\,\sin(45)\,\sin(55)\,\sin\theta_B\,\sin\theta_C \\
& + Z\,\sin(45)\,\sin(55)\,\sin\theta_B\,\sin\theta_C - d\,\sin(45)\,\sin(55)\,\sin\theta_B\,\sin\theta_C \\
& - G_{54,Y}\,\cos(45)\cos(55)^2\,\cos\theta_C\,\sin(45) - G_{54,Y}\cos(45)^2\,\cos(55)\,\cos\theta_C\,\sin\theta_B \\
& + G_{54,X}\,\cos(45)\cos(55)^2\,\sin(45)\,\sin\theta_C - G_{54,X}\cos(45)^2\,\cos(55)\,\sin\theta_B\,\sin\theta_C \\
& - G_{54,Y}\,\cos(55)\,\cos\theta_C\sin(45)^2\,\sin\theta_B + Y\,\cos(45)\cos(55)^2\,\cos\theta_C\,\sin(45) \\
& + Y\cos(45)^2\,\cos(55)\,\cos\theta_C\,\sin\theta_B - G_{54,X}\,\cos(55)\sin(45)^2\,\sin\theta_B\,\sin\theta_C \\
& - X\,\cos(45)\cos(55)^2\,\sin(45)\,\sin\theta_C + X\cos(45)^2\,\cos(55)\,\sin\theta_B\,\sin\theta_C \\
& + Y\,\cos(55)\,\cos\theta_C\sin(45)^2\,\sin\theta_B + X\,\cos(55)\sin(45)^2\,\sin\theta_B\,\sin\theta_C \\
& - G_{54,X}\cos(55)^2\,\cos\theta_B\,\cos\theta_C\sin(45)^2 + G_{54,Y}\cos(45)^2\cos(55)^2\,\cos\theta_B\,\sin\theta_C \\
& + X\cos(55)^2\,\cos\theta_B\,\cos\theta_C\sin(45)^2 - Y\cos(45)^2\cos(55)^2\,\cos\theta_B\,\sin\theta_C \\
& + Z\,\cos(45)\,\cos(55)\,\cos\theta_B\,\sin(55)\,\sin\theta_C + Z\,\cos(55)\,\cos\theta_B\,\cos\theta_C\,\sin(45)\,\sin(55) \\
& - d\,\cos(45)\,\cos(55)\,\cos\theta_B\,\sin(55)\,\sin\theta_C - d\,\cos(55)\,\cos\theta_B\,\cos\theta_C\,\sin(45)\,\sin(55) \\
& + G_{54,Y}\,\cos(45)\cos(55)^2\,\cos\theta_B\,\cos\theta_C\,\sin(45) - G_{54,X}\,\cos(45)\cos(55)^2\,\cos\theta_B\,\sin(45)\,\sin\theta_C \\
& - Y\,\cos(45)\cos(55)^2\,\cos\theta_B\,\cos\theta_C\,\sin(45) + X\,\cos(45)\cos(55)^2\,\cos\theta_B\,\sin(45)\,\sin\theta_C \\
& - G_{54,Z}\,\cos(45)\,\cos(55)\,\cos\theta_B\,\sin(55)\,\sin\theta_C - G_{54,Z}\,\cos(55)\,\cos\theta_B\,\cos\theta_C\,\sin(45)\,\sin(55)
\end{aligned}
\tag{A.33}
$$

$$
\begin{aligned}
y = {} & Y\cos(45)^2\cos\theta_C - G_{54,Y}\cos(45)^2\cos\theta_C - y_{Offset} - G_{54,X}\sin(45)^2\sin\theta_C + X\sin(45)^2\sin\theta_C \\
& + G_{54,X}\cos(55)^2\sin(45)^2\sin\theta_C - X\cos(55)^2\sin(45)^2\sin\theta_C + G_{54,X}\cos(45)\cos\theta_C\sin(45) \\
& + G_{54,Y}\cos(45)\sin(45)\sin\theta_C - X\cos(45)\cos\theta_C\sin(45) - Y\cos(45)\sin(45)\sin\theta_C \\
& - G_{54,X}\cos(45)^2\cos\theta_B\sin\theta_C - G_{54,Y}\cos\theta_B\cos\theta_C\sin(45)^2 + X\cos(45)^2\cos\theta_B\sin\theta_C \\
& + Y\cos\theta_B\cos\theta_C\sin(45)^2 + G_{54,Y}\cos(45)^2\cos(55)^2\cos\theta_C - Y\cos(45)^2\cos(55)^2\cos\theta_C \\
& - G_{54,X}\cos(45)\cos\theta_B\cos\theta_C\sin(45) - G_{54,Z}\cos(45)\cos(55)\cos\theta_C\sin(55) \\
& - G_{54,Y}\cos(45)\cos\theta_B\sin(45)\sin\theta_C + X\cos(45)\cos\theta_B\cos\theta_C\sin(45) \\
& + Z\cos(45)\cos(55)\cos\theta_C\sin(55) - d\cos(45)\cos(55)\cos\theta_C\sin(55) \\
& + G_{54,Z}\cos(55)\sin(45)\sin(55)\sin\theta_C + G_{54,Z}\cos(45)\sin(55)\sin\theta_B\sin\theta_C \\
& + G_{54,Z}\cos\theta_C\sin(45)\sin(55)\sin\theta_B + Y\cos(45)\cos\theta_B\sin(45)\sin\theta_C \\
& - Z\cos(55)\sin(45)\sin(55)\sin\theta_C - Z\cos(45)\sin(55)\sin\theta_B\sin\theta_C \\
& - Z\cos\theta_C\sin(45)\sin(55)\sin\theta_B + d\cos(55)\sin(45)\sin(55)\sin\theta_C \\
& + d\cos(45)\sin(55)\sin\theta_B\sin\theta_C + d\cos\theta_C\sin(45)\sin(55)\sin\theta_B \\
& - G_{54,X}\cos(45)\cos(55)^2\cos\theta_C\sin(45) + G_{54,X}\cos(45)^2\cos(55)\cos\theta_C\sin\theta_B \\
& - G_{54,Y}\cos(45)\cos(55)^2\sin(45)\sin\theta_C + G_{54,X}\cos(55)\cos\theta_C\sin(45)^2\sin\theta_B \\
& - G_{54,Y}\cos(45)^2\cos(55)\sin\theta_B\sin\theta_C + X\cos(45)\cos(55)^2\cos\theta_C\sin(45) \\
& - X\cos(45)^2\cos(55)\cos\theta_C\sin\theta_B - G_{54,Y}\cos(55)\sin(45)^2\sin\theta_B\sin\theta_C \\
& - X\cos(55)\cos\theta_C\sin(45)^2\sin\theta_B + Y\cos(45)\cos(55)^2\sin(45)\sin\theta_C \\
& + Y\cos(45)^2\cos(55)\sin\theta_B\sin\theta_C + Y\cos(55)\sin(45)^2\sin\theta_B\sin\theta_C \\
& - G_{54,Y}\cos(45)^2\cos(55)^2\cos\theta_B\cos\theta_C + Y\cos(45)^2\cos(55)^2\cos\theta_B\cos\theta_C \\
& - G_{54,X}\cos(55)^2\cos\theta_B\sin(45)^2\sin\theta_C + X\cos(55)^2\cos\theta_B\sin(45)^2\sin\theta_C \\
& - Z\cos(45)\cos(55)\cos\theta_B\cos\theta_C\sin(55) + d\cos(45)\cos(55)\cos\theta_B\cos\theta_C\sin(55) \\
& - G_{54,Z}\cos(55)\cos\theta_B\sin(45)\sin(55)\sin\theta_C + Z\cos(55)\cos\theta_B\sin(45)\sin(55)\sin\theta_C \\
& - d\cos(55)\cos\theta_B\sin(45)\sin(55)\sin\theta_C + G_{54,X}\cos(45)\cos(55)^2\cos\theta_B\cos\theta_C\sin(45) \\
& + G_{54,Y}\cos(45)\cos(55)^2\cos\theta_B\sin(45)\sin\theta_C - X\cos(45)\cos(55)^2\cos\theta_B\cos\theta_C\sin(45) \\
& - Y\cos(45)\cos(55)^2\cos\theta_B\sin(45)\sin\theta_C + G_{54,Z}\cos(45)\cos(55)\cos\theta_B\cos\theta_C\sin(55)
\end{aligned}
\tag{A.34}
$$

$$
\begin{aligned}
z = {} & d - z_{Offset} - G_{54,Z}\cos\theta_B + Z\cos\theta_B - d\cos\theta_B - G_{54,Z}\cos(55)^2 \\
& + Z\cos(55)^2 - d\cos(55)^2 + G_{54,Z}\cos(55)^2\cos\theta_B - Z\cos(55)^2\cos\theta_B \\
& + d\cos(55)^2\cos\theta_B - G_{54,Y}\cos(55)\cos(55)\sin(55) + G_{54,X}\cos(55)\sin(55)\sin(55) \\
& - G_{54,X}\cos(55)\sin(55)\sin\theta_B + Y\cos(55)\cos(55)\sin(55) - G_{54,Y}\sin(55)\sin(55)\sin\theta_B \\
& - X\cos(55)\sin(55)\sin(55) + X\cos(55)\sin(55)\sin\theta_B + Y\sin(55)\sin(55)\sin\theta_B \\
& + G_{54,Y}\cos(55)\cos(55)\cos\theta_B\sin(55) - G_{54,X}\cos(55)\cos\theta_B\sin(55)\sin(55) \\
& - Y\cos(55)\cos(55)\cos\theta_B\sin(55) + X\cos(55)\cos\theta_B\sin(55)\sin(55)
\end{aligned}
\tag{A.35}
$$

A.4 Tool Position - Machine Coordinate System

$$
\begin{aligned}
&X \\
&= G_{54,X} - z\ (\cos(45)\ \sin(55)\ \sin\theta_B - \cos(55)\ \sin(45)\ \sin(55) + \cos(55)\ \cos\theta_B\ \sin(45)\ \sin(55)) \\
&- y\ \big(-\cos\theta_C\ \sin\theta_B \cos(45)^2\ \cos(55) + \cos\theta_B\ \sin\theta_C \cos(45)^2 \\
&\qquad - \cos\theta_B\ \cos\theta_C\ \cos(45)\cos(55)^2\ \sin(45) - \cos\theta_C\ \cos(45)\ \sin(45)\sin(55)^2 \\
&\qquad + \cos\theta_B\ \cos\theta_C\ \cos(45)\ \sin(45) + \cos\theta_B\ \sin\theta_C \cos(55)^2 \sin(45)^2 \\
&\qquad - \cos\theta_C\ \sin\theta_B\ \cos(55)\sin(45)^2 + \sin\theta_C \sin(45)^2 \sin(55)^2\big) \\
&- x\ \big(\sin\theta_B\ \sin\theta_C \cos(45)^2\ \cos(55) + \cos\theta_B\ \cos\theta_C \cos(45)^2 \\
&\qquad + \cos\theta_B\ \sin\theta_C\ \cos(45)\cos(55)^2\ \sin(45) + \sin\theta_C\ \cos(45)\ \sin(45)\sin(55)^2 \\
&\qquad - \cos\theta_B\ \sin\theta_C\ \cos(45)\ \sin(45) + \cos\theta_B\ \cos\theta_C \cos(55)^2 \sin(45)^2 \\
&\quad + \sin\theta_B\ \sin\theta_C\ \cos(55)\sin(45)^2 + \cos\theta_C \sin(45)^2 \sin(55)^2\big) - x_{\text{offset}}\ \cos\theta_C \sin(45)^2 \sin(55)^2 \\
&- y_{\text{offset}} \sin(45)^2 \sin(55)^2\ \sin\theta_C - d\ \cos(55)\ \sin(45)\ \sin(55) + d\ \cos(45)\ \sin(55)\ \sin\theta_B \\
&+ z_{\text{offset}}\ \cos(55)\ \sin(45)\ \sin(55) - z_{\text{offset}}\ \cos(45)\ \sin(55)\ \sin\theta_B - x_{\text{offset}} \cos(45)^2\ \cos\theta_B\ \cos\theta_C \\
&- y_{\text{offset}} \cos(45)^2\ \cos\theta_B\ \sin\theta_C - x_{\text{offset}} \cos(55)^2\ \cos\theta_B\ \cos\theta_C \sin(45)^2 \\
&- y_{\text{offset}} \cos(55)^2\ \cos\theta_B \sin(45)^2\ \sin\theta_C + d\ \cos(55)\ \cos\theta_B\ \sin(45)\ \sin(55) \\
&- y_{\text{offset}}\ \cos(45)\ \cos\theta_B\ \cos\theta_C\ \sin(45) + x_{\text{offset}}\ \cos(45)\ \cos\theta_B\ \sin(45)\ \sin\theta_C \\
&- z_{\text{offset}}\ \cos(55)\ \cos\theta_B\ \sin(45)\ \sin(55) + y_{\text{offset}} \cos(45)^2\ \cos(55)\ \cos\theta_C\ \sin\theta_B \\
&- x_{\text{offset}} \cos(45)^2\ \cos(55)\ \sin\theta_B\ \sin\theta_C + y_{\text{offset}}\ \cos(45)\ \cos\theta_C\ \sin(45)\sin(55)^2 \\
&+ y_{\text{offset}}\ \cos(55)\ \cos\theta_C \sin(45)^2\ \sin\theta_B - x_{\text{offset}}\ \cos(45)\ \sin(45)\sin(55)^2\ \sin\theta_C \\
&- x_{\text{offset}}\ \cos(55)\sin(45)^2\ \sin\theta_B\ \sin\theta_C + y_{\text{offset}}\ \cos(45)\cos(55)^2\ \cos\theta_B\ \cos\theta_C\ \sin(45) \\
&- x_{\text{offset}}\ \cos(45)\cos(55)^2\ \cos\theta_B\ \sin(45)\ \sin\theta_C
\end{aligned}
\tag{A.36}
$$

$$
\begin{aligned}
&Y \\
&= G_{54,Y} - z\ (\sin(45)\ \sin(55)\ \sin\theta_B + \cos(45)\ \cos(55)\ \sin(55) - \cos(45)\ \cos(55)\ \cos\theta_B\ \sin(55)) \\
&+ x\ \big(\cos\theta_B\ \sin\theta_C \cos(45)^2 \cos(55)^2 - \cos\theta_C\ \sin\theta_B \cos(45)^2\ \cos(55) + \sin\theta_C \cos(45)^2 \sin(55)^2 \\
&\qquad + \cos\theta_B\ \cos\theta_C\ \cos(45)\cos(55)^2\ \sin(45) + \cos\theta_C\ \cos(45)\ \sin(45)\sin(55)^2 \\
&\qquad - \cos\theta_B\ \cos\theta_C\ \cos(45)\ \sin(45) - \cos\theta_C\ \sin\theta_B\ \cos(55)\sin(45)^2 + \cos\theta_B\ \sin\theta_C \sin(45)^2\big) \\
&- y\ \big(\cos\theta_B\ \cos\theta_C \cos(45)^2 \cos(55)^2 + \sin\theta_B\ \sin\theta_C \cos(45)^2\ \cos(55) + \cos\theta_C \cos(45)^2 \sin(55)^2 \\
&\qquad - \cos\theta_B\ \sin\theta_C\ \cos(45)\cos(55)^2\ \sin(45) - \sin\theta_C\ \cos(45)\ \sin(45)\sin(55)^2 \\
&\qquad + \cos\theta_B\ \sin\theta_C\ \cos(45)\ \sin(45) + \sin\theta_B\ \sin\theta_C\ \cos(55)\sin(45)^2 + \cos\theta_B\ \cos\theta_C \sin(45)^2\big) \\
&- y_{\text{offset}} \cos(45)^2\ \cos\theta_C \sin(55)^2 + x_{\text{offset}} \cos(45)^2 \sin(55)^2\ \sin\theta_C \\
&+ d\ \cos(45)\ \cos(55)\ \sin(55) - z_{\text{offset}}\ \cos(45)\ \cos(55)\ \sin(55) + d\ \sin(45)\ \sin(55)\ \sin\theta_B \\
&- z_{\text{offset}}\ \sin(45)\ \sin(55)\ \sin\theta_B - y_{\text{offset}}\ \cos\theta_B\ \cos\theta_C \sin(45)^2 + x_{\text{offset}}\ \cos\theta_B \sin(45)^2\ \sin\theta_C \\
&+ x_{\text{offset}} \cos(45)^2 \cos(55)^2\ \cos\theta_B\ \sin\theta_C - d\ \cos(45)\ \cos(55)\ \cos\theta_B\ \sin(55) \\
&- x_{\text{offset}}\ \cos(45)\ \cos\theta_B\ \cos\theta_C\ \sin(45) + z_{\text{offset}}\ \cos(45)\ \cos(55)\ \cos\theta_B\ \sin(55) \\
&- y_{\text{offset}}\ \cos(45)\ \cos\theta_B\ \sin(45)\ \sin\theta_C - x_{\text{offset}} \cos(45)^2\ \cos(55)\ \cos\theta_C\ \sin\theta_B \\
&+ x_{\text{offset}}\ \cos(45)\ \cos\theta_C\ \sin(45)\sin(55)^2 - x_{\text{offset}}\ \cos(55)\ \cos\theta_C \sin(45)^2\ \sin\theta_B \\
&- y_{\text{offset}} \cos(45)^2\ \cos(55)\ \sin\theta_B\ \sin\theta_C + y_{\text{offset}}\ \cos(45)\ \sin(45)\sin(55)^2\ \sin\theta_C \\
&- y_{\text{offset}}\ \cos(55)\sin(45)^2\ \sin\theta_B\ \sin\theta_C - y_{\text{offset}} \cos(45)^2 \cos(55)^2\ \cos\theta_B\ \cos\theta_C \\
&+ x_{\text{offset}}\ \cos(45)\cos(55)^2\ \cos\theta_B\ \cos\theta_C\ \sin(45) + y_{\text{offset}}\ \cos(45)\cos(55)^2\ \cos\theta_B\ \sin(45)\ \sin\theta_C
\end{aligned}
\tag{A.37}
$$

$$
\begin{aligned}
Z \\
&= G_{54,Z} + d + z_{\text{offset}} \cos(55)^2 - x\ (\cos(45)\ \cos(55)\ \sin(55)\ \sin\theta_C + \cos(55)\ \cos\theta_C\ \sin(45)\ \sin(55) \\
&\qquad + \cos(45)\ \cos\theta_C\ \sin(55)\ \sin\theta_B - \sin(45)\ \sin(55)\ \sin\theta_B\ \sin\theta_C \\
&\qquad - \cos(45)\ \cos(55)\ \cos\theta_B\ \sin(55)\ \sin\theta_C - \cos(55)\ \cos\theta_B\ \cos\theta_C\ \sin(45)\ \sin(55)) \\
&\quad - y\ (\cos(55)\ \sin(45)\ \sin(55)\ \sin\theta_C - \cos(45)\ \cos(55)\ \cos\theta_C\ \sin(55) \\
&\qquad + \cos(45)\ \sin(55)\ \sin\theta_B\ \sin\theta_C + \cos\theta_C\ \sin(45)\ \sin(55)\ \sin\theta_B \\
&\qquad + \cos(45)\ \cos(55)\ \cos\theta_B\ \cos\theta_C\ \sin(55) - \cos(55)\ \cos\theta_B\ \sin(45)\ \sin(55)\ \sin\theta_C) \\
&\quad + z\ \left(\cos(55)^2 + \cos\theta_B \sin(55)^2\right) - d\cos(55)^2 - d\ \cos\theta_B \sin(55)^2 + z_{\text{offset}}\ \cos\theta_B \sin(55)^2 \\
&\qquad + y_{\text{offset}}\ \cos(45)\ \cos(55)\ \cos\theta_C\ \sin(55) - x_{\text{offset}}\ \cos(45)\ \cos(55)\ \sin(55)\ \sin\theta_C \\
&\qquad - x_{\text{offset}}\ \cos(55)\ \cos\theta_C\ \sin(45)\ \sin(55) - x_{\text{offset}}\ \cos(45)\ \cos\theta_C\ \sin(55)\ \sin\theta_B \\
&\qquad - y_{\text{offset}}\ \cos(55)\ \sin(45)\ \sin(55)\ \sin\theta_C - y_{\text{offset}}\ \cos(45)\ \sin(55)\ \sin\theta_B\ \sin\theta_C \\
&\qquad - y_{\text{offset}}\ \cos\theta_C\ \sin(45)\ \sin(55)\ \sin\theta_B + x_{\text{offset}}\ \sin(45)\ \sin(55)\ \sin\theta_B\ \sin\theta_C \\
&\qquad - y_{\text{offset}}\ \cos(45)\ \cos(55)\ \cos\theta_B\ \cos\theta_C\ \sin(55) + x_{\text{offset}}\ \cos(45)\ \cos(55)\ \cos\theta_B\ \sin(55)\ \sin\theta_C \\
&\qquad + x_{\text{offset}}\ \cos(55)\ \cos\theta_B\ \cos\theta_C\ \sin(45)\ \sin(55) + y_{\text{offset}}\ \cos(55)\ \cos\theta_B\ \sin(45)\ \sin(55)\ \sin\theta_C
\end{aligned}
\tag{A.38}
$$

Bibliography

[1] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of Robots by learning," *J. Robot. Syst.*, vol. 1, no. 2, pp. 123–140, 1984. [Online]. Available: http://doi.wiley.com/10.1002/rob.4620010203

[2] R. Ward, O. Ozkirimli, and B. Jones, "Increasing Part Geometric Accuracy in High Speed Machining using Cascade Iterative Learning Control," *Procedia CIRP*, vol. 101, pp. 298–301, jan 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2212827121007150 https://doi.org/10.1016/j.procir.2020.10.006

[3] Y. Altintas, "Virtual High Performance Machining," *7th HPC 2016 – CIRP Conf. High Perform. Cut.*, vol. 46, pp. 372–378, 2016.

[4] M. Armendia, M. Ghassempouri, E. Ozturk, and F. Peysson, *Twin-Control, A Digital Twin Approach to Improve Machine Tools Lifecycle*, 1st ed. Springer, 2019.

[5] S. E. Layegh K., H. Erdim, and I. Lazoglu, "Offline force control and feedrate scheduling for complex free form surfaces in 5-axis milling," *Procedia CIRP*, vol. 1, no. 1, pp. 96–101, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.procir.2012.04.015

[6] L. Berglind, D. Plakhotnik, and E. Ozturk, "Discrete Cutting Force Model for 5-Axis Milling with Arbitrary Engagement and Feed Direction," *Procedia CIRP*, vol. 58, no. Cl, pp. 445–450, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.procir.2017.03.250

[7] Z. Nie, R. Lynn, T. Tucker, and T. Kurfess, "Voxel-based analysis and modeling of MRR computational accuracy in milling process," *CIRP J. Manuf. Sci. Technol.*, vol. 27, pp. 78–92, 2019.

[8] B. Sencer and Y. Altintas, "Modeling and control of contouring errors for five-axis machine tools-Part II: Precision contour controller design," *J. Manuf. Sci. Eng. Trans. ASME*, vol. 131, no. 3, pp. 0 310 071–03 100 710, 2009.

[9] Y. Altintas, P. Kersting, D. Biermann, E. Budak, B. Denkena, and I. Lazoglu, "Virtual process systems for part machining operations," *CIRP Ann.*, vol. 63, no. 2, pp. 585–605, jan 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cirp.2014.05.007 https://www.sciencedirect.com/science/article/pii/S0007850614001899

[10] Y. Altintas, *Manufacturing Automation*. Cambridge: Cambridge University Press, 2011. [Online]. Available: http://ebooks.cambridge.org/ref/id/CBO9780511843723

[11] Y. K. Choi and A. Banerjee, "Tool path generation and tolerance analysis for free-form surfaces," *Int. J. Mach. Tools Manuf.*, vol. 47, no. 3-4, pp. 689–696, 2007.

[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, no. 7-8, pp. 733–745, 2008.

[13] 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, pp. 1323–1345, 2001.

[14] X. Beudaert, S. Lavernhe, and C. Tournier, "Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path," *Int. J. Mach. Tools Manuf.*, vol. 57, pp. 73–82, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.ijmachtools.2012.02.005

[15] HEIDENHAIN, "TNC 640 HSCI The Contouring Control for Milling-Turning and Machining Centers," *Inf. Mach. Tool Build. Man.*, no. June, 2017.

[16] S. Y. Jeong, Y. J. Choi, P. G. Park, and S. G. Choi, "Jerk limited velocity profile generation for high speed industrial robot trajectories," *IFAC Proc. Vol.*, vol. 16, pp. 595–600, 2005.

[17] P. J. Barre, R. Bearee, P. Borne, and E. Dumetz, "Influence of a jerk controlled movement law on the vibratory behaviour of high-dynamics systems," *J. Intell. Robot. Syst. Theory Appl.*, vol. 42, no. 3, pp. 275–293, 2005.

[18] Y. Altintas and S. Tulsyan, "Prediction of part machining cycle times via virtual CNC," *CIRP Ann. - Manuf. Technol.*, vol. 64, no. 1, pp. 361–364, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.cirp.2015.04.100

[19] 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, 2006.

[20] S. Tulsyan and Y. Altintas, "Cycle time prediction for milling free-form surfaces by considering dynamics of CNC systems."

[21] S. Tajima, B. Sencer, and E. Shamoto, "Accurate interpolation of machining tool-paths based on FIR filtering," *Precis. Eng.*, vol. 52, no. December 2017, pp. 332–344, 2018. [Online]. Available: http://dx.doi.org/10.1016/j.precisioneng.2018.01.016

[22] Heidenhain, "Machining Accuracy of Machine Tools," *Heidenhain White Pap.*, no. September, 2011.

[23] L. Vinet and A. Zhedanov, "A 'missing' family of classical orthogonal polynomials," 2011.

[24] Siemens, "Sinumerik One Dynamics, software version NCU-SW 6.14," 2020.

[25] S. Tajima and B. Sencer, "Accurate real-time interpolation of 5-axis tool-paths with local corner smoothing," *Int. J. Mach. Tools Manuf.*, vol. 142, no. February, pp. 1–15, 2019. [Online]. Available: https://doi.org/10.1016/j.ijmachtools.2019.04.005

[26] ——, "Real-time trajectory generation for 5-axis machine tools with singularity avoidance," *CIRP Ann.*, vol. 69, no. 1, pp. 349–352, 2020. [Online]. Available: https://doi.org/10.1016/j.cirp.2020.04.050

[27] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Control Eng. Pract.*, vol. 13, no. 2, pp. 145–157, 2005.

[28] L. Biagiotti and C. Melchiorri, "FIR filters for online trajectory planning with time- and frequency-domain specifications," *Control Eng. Pract.*, vol. 20, no. 12, pp. 1385–1399, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.conengprac.2012.08.005

[29] E. R. Ziegel, H. Blinchikoff, and A. Zverev, *Filtering in the Time and Frequency Domains*, 1988, vol. 30, no. 2.

[30] R. Ur-Rehman, C. Richterich, K. Arntz, and F. Klocke, "Numerical techniques for CAM strategies for machining of mould and die," in *Proc. 36th Int. MATADOR Conf.*, S. Hinduja and L. Li, Eds., vol. 2010-Janua. London: Springer London, 2010, pp. 259–263.

[31] Siemens, "Siemens NX," pp. 11–13, 2019. [Online]. Available: https://www.plm.automation.siemens.com/global/es/products/nx/

[32] C. Geng, Y. Wu, and J. Qiu, "Analysis of Nonlinear Error Caused by Motions of Rotation Axes for Five-Axis Machine Tools with Orthogonal Configuration," *Math. Probl. Eng.*, vol. 2018, 2018.

[33] K. Zhang, L. Zhang, and Y. Yan, "Single spherical angle linear interpolation for the control of non-linearity errors in five-axis flank milling," *Int. J. Adv. Manuf. Technol.*, vol. 87, no. 9-12, pp. 3289–3299, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00170-016-8720-3

[34] K. Erkorkmaz and Y. Altintas, "Quintic Spline Interpolation With Minimal Feed Fluctuation," *J. Manuf. Sci. Eng.*, vol. 127, no. 2, p. 339, 2005.

[35] C. H. Yeung, Y. Altintas, and K. Erkorkmaz, "Virtual CNC system. Part I. System architecture," *Int. J. Mach. Tools Manuf.*, vol. 46, no. 10, pp. 1107–1123, 2006.

[36] 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, no. 9-12, pp. 1977–1992, 2015.

[37] C. A. Ernesto and R. T. Farouki, "High-speed cornering by CNC machines under prescribed bounds on axis accelerations and toolpath contour error," *Int. J. Adv. Manuf. Technol.*, vol. 58, no. 1-4, pp. 327–338, 2012.

[38] M. T. Lin, M. S. Tsai, and H. T. Yau, "Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm," *Int. J. Mach. Tools Manuf.*, vol. 47, no. 15, pp. 2246–2262, 2007.

[39] H. Zhao, L. M. Zhu, and H. Ding, "A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segments," *Int. J. Mach. Tools Manuf.*, vol. 65, pp. 88–98, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.ijmachtools.2012.10.005

[40] X. Beudaert, S. Lavernhe, and C. Tournier, "5-Axis Local Corner Rounding of Linear Tool Path Discontinuities," *Int. J. Mach. Tools Manuf.*, vol. 73, pp. 9–16, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.ijmachtools.2013.05.008

[41] S. Tulsyan and Y. Altintas, "Local toolpath smoothing for five-axis machine tools," *Int. J. Mach. Tools Manuf.*, vol. 96, pp. 15–26, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.ijmachtools.2015.04.014

[42] S. Sun and Y. Altintas, "A G3 continuous tool path smoothing method for 5-axis CNC machining," *CIRP J. Manuf. Sci. Technol.*, vol. 32, pp. 529–549, 2021. [Online]. Available: https://doi.org/10.1016/j.cirpj.2020.11.002

[43] 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, no. May 2017, pp. 67–79, 2018.

[44] Q. Bi, J. Shi, Y. Wang, L. Zhu, and H. Ding, "Analytical curvature-continuous dual-Bézier corner transition for five-axis linear tool path," *Int. J. Mach. Tools Manuf.*, vol. 91, pp. 96–108, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.ijmachtools.2015.02.002

[45] J. Yang and A. Yuen, "An analytical local corner smoothing algorithm for five-axis CNC machining," *Int. J. Mach. Tools Manuf.*, vol. 123, no. July, pp. 22–35, 2017.

[46] S. Tajima and B. Sencer, "Global tool-path smoothing for CNC machine tools with uninterrupted acceleration," *Int. J. Mach. Tools Manuf.*, vol. 121, no. November 2016, pp. 81–95, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.ijmachtools.2017.03.002

[47] D. Li, L. Zhang, L. Yang, and J. Mao, "Accurate interpolation and contouring control algorithm based on FIR filters for the corner transition," *Int. J. Adv. Manuf. Technol.*, vol. 109, no. 7-8, pp. 1775–1788, 2020.

[48] R. Ward, B. Sencer, B. Jones, and E. Ozturk, "Accurate prediction of machining feedrate and cycle times considering interpolator dynamics," *Int. J. Adv. Manuf. Technol.*, vol. 116, no. 1-2, pp. 417–438, sep 2021.

[49] Y. Liu, M. Wan, X. B. Qin, Q. B. Xiao, and W. H. Zhang, "FIR filter-based continuous interpolation of G01 commands with bounded axial and tangential kinematics in industrial five-axis machine tools," *Int. J. Mech. Sci.*, vol. 169, no. November 2019, p. 105325, 2020. [Online]. Available: https://doi.org/10.1016/j.ijmecsci.2019.105325

[50] Y. Jiang, J. Han, L. Xia, L. Lu, X. Tian, and H. Liu, "A decoupled five-axis local smoothing interpolation method to achieve continuous acceleration of tool axis," *Int. J. Adv. Manuf. Technol.*, vol. 111, no. 1-2, pp. 449–470, 2020.

[51] S. Tajima, B. Sencer, H. Yoshioka, and H. Shinno, "Smooth path blending for 5-axis machine tools," in *JSME 2020 Conf. Lead. Edge Manuf. Process. LEMP 2020*, 2020, pp. 2–6.

[52] X. Beudaert, P. Y. Pechard, and C. Tournier, "5-Axis Tool Path Smoothing Based on Drive Constraints," *Int. J. Mach. Tools Manuf.*, vol. 51, no. 12, pp. 958–965, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.ijmachtools.2011.08.014

[53] X. Huang, F. Zhao, T. Tao, and X. Mei, "A Novel Local Smoothing Method for Five-Axis Machining with Time-Synchronization Feedrate Scheduling," *IEEE Access*, vol. 8, pp. 89 185–89 204, 2020.

[54] Y. Zhang, T. Wang, P. Peng, J. Dong, L. Cao, and C. Tian, "Feedrate blending method for five-axis linear tool path under geometric and kinematic constraints," *Int. J. Mech. Sci.*, vol. 195, no. December 2020, 2021.

[55] Y. H. Jung, D. W. Lee, J. S. Kim, and H. S. Mok, "NC post-processor for 5-axis milling machine of table-rotating/tilting type," *J. Mater. Process. Technol.*, vol. 130-131, pp. 641–646, dec 2002.

[56] M. C. Ho, Y. R. Hwang, and C. H. Hu, "Five-axis tool orientation smoothing using quaternion interpolation algorithm," *Int. J. Mach. Tools Manuf.*, vol. 43, no. 12, pp. 1259–1267, 2003.

[57] G. A. Jennings, *Spherical Geometry*. New York, NY: Springer New York, 1994, pp. 43–82. [Online]. Available: https://doi.org/10.1007/978-1-4612-0855-6_2

[58] B. Jousselin, Y. Quinsat, and C. Tournier, "Tool axis adjustment for 5-axis roughing operations," *CIRP J. Manuf. Sci. Technol.*, vol. 35, pp. 615–623, 2021.

[59] D. Li, W. Zhang, W. Zhou, T. Shang, and J. Fleischer, "Dual NURBS Path Smoothing for 5-Axis Linear Path of Flank Milling," *Int. J. Precis. Eng. Manuf.*, vol. 19, no. 12, pp. 1811–1820, 2018.

[60] J. Zhang, L. Zhang, K. Zhang, and J. Mao, "Double NURBS trajectory generation and synchronous interpolation for five-axis machining based on dual quaternion algorithm," *Int. J. Adv. Manuf. Technol.*, vol. 83, no. 9-12, pp. 2015–2025, 2016.

[61] R. Ward, B. Sencer, B. Jones, and E. Ozturk, "Five-axis trajectory generation considering synchronisation and nonlinear interpolation error," *J. Manuf. Sci. Eng.*, 2022.

[62] D. N. Song, J. W. Ma, Y. G. Zhong, D. Xiao, J. J. Yao, and C. Zhou, "A fully real-time spline interpolation algorithm with axial jerk constraint based on FIR filtering," *Int. J. Adv. Manuf. Technol.*, vol. 113, no. 7-8, pp. 1873–1886, 2021.

[63] M. Borgerding, "Turning overlap-save into a multiband mixing, downsampling filter bank," *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 158–161, 2006.

[64] S. W. Smith, *Moving Average Filters*, S. W.-D. S. P. Smith, Ed. Boston: Newnes, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780750674447500522

[65] R. Oshana, *DSP Software Development Techniques for Embedded and Real-Time Systems*. Elsevier Science & Technology, 2006.

[66] K. Sørby, "Inverse kinematics of five-axis machines near singular configurations," *Int. J. Mach. Tools Manuf.*, vol. 47, no. 2, pp. 299–306, 2007.