Smooth and Time-Optimal Trajectory Generation for High Speed Machine Tools

by

Michele Mei-Ting Heng

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Mechanical Engineering

Waterloo, Ontario, Canada, 2008

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In machining complex dies, molds, aerospace and automotive parts, or biomedical components, it is crucial to minimize the cycle time, which reduces costs, while preserving the quality and tolerance integrity of the part being produced. To meet the demands for high quality finishes and low production costs in machining parts with complex geometry, computer numerical control (CNC) machine tools must be equipped with spline interpolation, feedrate modulation, and feedrate optimization capabilities. This thesis presents the development of novel trajectory generation algorithms for Non Uniform Rational B-Spline (NURBS) toolpaths that can be implemented on new low-cost CNC's, as well as, in conjunction with existing CNC's. In order to minimize feedrate fluctuations during the interpolation of NURBS toolpaths, the concept of the *feed correction polynomial* is applied. Feedrate fluctuations are reduced from around 40 % for natural interpolation to 0.1 % for interpolation with feed correction. Excessive acceleration and jerk in the axes are also avoided. To generate jerk-limited feed motion profiles for long segmented toolpaths, a generalized framework for feedrate modulation, based on the S-curve function, is presented. Kinematic compatibility conditions are derived to ensure that the position, velocity, and acceleration profiles are continuous and that the jerk is limited in all axes. This framework serves as the foundation for the proposed heuristic feedrate optimization strategy in this thesis. Using analytically derived kinematic compatibility equations and an efficient bisection search algorithm, the command feedrate for each segment is maximized. Feasible solutions must satisfy the optimization constraints on the velocity, control signal (i.e. actuation torque), and jerk in each axis throughout the trajectory. The maximized feedrates are used to generate near-optimal feed profiles that have shorter cycle times, approximately 13-26% faster than the feed profiles obtained using the worst-case curvature approach, which is widely used in industrial CNC interpolators. The effectiveness of the NURBS interpolation, feedrate modulation and feedrate optimization techniques has been verified in 3-axis machining experiments of a biomedical implant.

Acknowledgements

This research was sponsored by the Natural Science and Engineering Research Council (NSERC), the Ontario Centres of Excellence (OCE) and the Canada Foundation for Innovation (CFI) granting agencies and was completed during the tenure of an NSERC Canada Graduate Scholarship.

I gratefully acknowledge the assistance of Mr. R. Wagner and Mr. A. Rouzrokh in the machining experiments. I would also like to acknowledge Mr. W. Wong and Mr. A. Kamalzadeh for their kind answers in responding to any questions that I posed to them. The implant design was developed and provided by Prof. M. Papini, Mr. A. Dudi, Prof. P. Zalzal, and Prof. R.M. Pilliar.

Valuable feedback on the scientific content and presentation of this thesis has been provided by Prof. J. Huissoon and Prof. R. Fraser. I would like to thank them for taking the time to read this thesis in its entirety.

Lastly, I sincerely acknowledge the time and commitment that has been provided by my supervisor, Prof. K. Erkorkmaz. He has been a guide, mentor, teacher, and supporter. Under his coaching, I have gained a great deal of knowledge and have been able to create new knowledge through my research.

Dedication

To my mom and dad

Table of Contents

AUTHOR'S DECLARATION ii

Abstract iii

Acknowledgements iv

Dedication v

Table of Contents vi

List of Figures ix

List of Tables xi

Nomenclature xii

Chapter 1 Introduction 1

1.1 Introduction 1

Chapter 2 Literature Review 4

2.1 Introduction 4

2.2 Non Uniform Rational B-Splines (NURBS) Toolpaths 5

2.3 Toolpath Parameterization 7

2.4 Spline Interpolation 10

2.5 Feed Generation 13

2.6 Feed Optimization 15

2.7 Conclusions 17

Chapter 3 NURBS Trajectory Generation 18

3.1 Introduction 18

3.2 NURBS Toolpath Parameterization 18

3.3 NURBS Toolpath Interpolation 32

3.3.1 Segment Arc-length Calculation 32

3.3.2 Feed Correction Polynomial 34

3.4 Simulation Results 39

3.5 Conclusions 44

Chapter 4 Jerk Limited Feedrate Modulation 45

4.1 Introduction 45

4.2 Feed Profile Formulation 46

4.3 Kinematic Compatibility Conditions 56

4.4 Implementation Details 57

4.4.1 Look-ahead for Long Toolpaths 59

4.5 Experimental Results 62

4.6 Conclusions 67

Chapter 5 Feedrate Optimization 68

5.1 Introduction 68

5.2 Problem Formulation ... 68

5.3 Solution Methodology ... 71

5.3.1 Worst-case Technique ... 71

5.3.2 Heuristic Strategy ... 75

5.4 Experimental Results ... 86

5.5 Conclusions ... 91

Chapter 6 Conclusions and Future Work ... 92

References ... 95

Appendix A Non Uniform Rational B-Spline (NURBS) Format ... 99

Appendix B Kinematic Compatibility Derivations ... 105

List of Figures

Figure 2-1. Overview of toolpath command generation as developed in this thesis. The CAD/CAM system handles toolpath parameterization, arc length calculation and feed correction polynomial fitting, while the CNC controller performs feed generation, feed optimization and trajectory interpolation. 5

Figure 2-2. NURBS curve representation. [5] 7

Figure 2-3. Shape modification of a NURBS curve using control points and weights. [5]...... 8

Figure 3-1. Optimal placement of control points to fit multiple NURBS segments to data points. 19

Figure 3-2. Evaluation of zero (*p*=0), first (*p*=1) and second (*p*=2) degree B-spline basis functions. 20

Figure 3-3. Division of data points and computation of spline parameter values for each data point (*p* = 3, *n* = 4). 22

Figure 3-4. Sample calculation of the jth knot in a knot vector (*p* = 3, *n* = 5). 23

Figure 3-5. Effect of tangent beta-constraint value, β1, on curve fitting............................... 27

Figure 3-6. Rule of thumb to calculate tangent beta value, β1. 28

Figure 3-7. Evaluation of the first derivative B-spline basis function (*p* = 3). 29

Figure 3-8. a) Feed correction polynomial fitting and comparison of its analytical and numerical derivatives. There are 8 polynomials. b) Mean squared error (MSE) of curve fit for each polynomial. 34

Figure 3-9. 17-segment NURBS toolpath.. 39

Figure 3-10. Toolpath 1 - comparison of NURBS interpolation without and with feed correction. 41

Figure 3-11. Toolpath 2 - comparison of resultant feedrate and axis kinematic profiles without and with feed correction. 43

Figure 4-1. Feed modulation demonstrated with a 7-segment spline toolpath example. 45

Figure 4-2. Initial (*1*st) segment profile. 47

Figure 4-3. Middle (*k*th) segment profile. 47

Figure 4-4. Final (*N*th) segment profile. 47

Figure 4-5. Implementation of feed modulation strategy with a look-ahead window. 58

Figure 4-6. Bisection search algorithm for a feasible feed. 59

Figure 4-7. Deceleration from maximum feed to rest. 60

Figure 4-8. Experimental setup (4' x 8' router table). 63

Figure 4-9. Biomedical implant model to machined part. 64

Figure 4-10. Kinematic profiles and controlled contouring results for the sample NURBS toolpath. 66

Figure 5-1. Comparison of worst-case feed optimization [31] to the proposed heuristic strategy. 75

Figure 5-2. Proposed heuristic feed optimization technique (demonstrated with an example). 77

Figure 5-3. Bisection search algorithm to find an optimized feasible feed. 78

Figure 5-4. Constraint evaluation points based on the variation of the toolpath geometry. 82

Figure 5-5. Benchmark contour - 88-segment quintic spline toolpath [9]. 86

Figure 5-6. Optimized feedrate profiles using worst-case [31], gradient-based [32], and heuristic (proposed) optimization strategies 87

Figure 5-7. Comparison of tracking performance for different optimization strategies. 88

Figure 5-8. Optimized feedrate profiles of a sample NURBS toolpath. 89

Figure 5-9. Tracking performance of contour machining for a biomedical implant. 90

List of Tables

Table 3-1. Segmentation of data points. 22

Table 3-2. Toolpath 1 - comparison of min/max axis acceleration and jerk values. 41

Table 3-3. Toolpath 2 - comparison of min/max axis acceleration and jerk values. 42

Table 4-1. Identified control signal equivalent parameters of the experimental setup. 63

Table 4-2. Tuned sliding mode control (SMC) parameters for the experimental setup. 64

Table 5-1. Sample calculation of the number of bisection operations for a range defined by $F_{k,\mathrm{min}}$=0 and $F_{k,\mathrm{max}}$=100 mm/s. 85

Table 5-2. Computational time for benchmark toolpath feed optimization........................... 88

Table 5-3. Cycle time comparison for implant surface... 90

Nomenclature

Δ_k	feed increment
Δs	change in arc length
$\mathbf{\Lambda}$	Lagrange multiplers
Φ	regressor matrix
$\alpha_0, \alpha_1, \alpha_2, ..., \alpha_7$	normalized coefficients of the feed correction polynomial
$\beta 1, \beta 2$	beta values for geometric continuity constraints
δ_w	feedrate increment
ε	arc length calculation tolerance
ε_{MSE}	mean square error curve fitting tolerance
$\boldsymbol{\varphi}$	regressor matrix
$\boldsymbol{\varphi}_k$	regressor matrix for the *k*th spline segment data points
$\boldsymbol{\theta}$	parameter vector
$\boldsymbol{\sigma}$	normalized arc displacement values
τ	relative time parameter
$\boldsymbol{\xi}$	constraint vector
$\boldsymbol{\xi}_x$	x-axis coordinates constraint vector
$A_0, A_1, A_2, ..., A_7$	coefficients of the feed correction polynomial
A_k	tangential acceleration from *k-1*th to the *k*th spline segment
$A_{\max}$	maximum tangential acceleration
$\mathbf{B}$	diagonal matrix of control signal equivalent damping for x, y, and z axes
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CNC	Computer Numerical Control
$\mathbf{C}$	point on a NURBS curve
$\dot{\mathbf{C}}$	velocity vector on a NURBS curve
$\ddot{\mathbf{C}}$	acceleration vector on a NURBS curve
$\dddot{\mathbf{C}}$	jerk vector on aNURBS curve
$\mathbf{C}_k$	point on the *k*th NURBS curve
$\mathbf{C}_k^{(1)}$	first parametric derivative vector on the *k*th NURBS curve
$\mathbf{C}_k^{(2)}$	second parametric derivative vector on the *k*th NURBS curve
$\mathbf{C}_s$	first geometric derivative (tangent) vector on a NURBS curve
$\mathbf{C}_{ss}$	second geometric derivative (curvature) vector on a NURBS curve
$\mathbf{C}_{sss}$	third geometric derivative vector on a NURBS curve
$C^n : C^0, C^1, C^2$	parametric continuity: zero, first, and second order
$\mathbf{D}_{coul}$	diagonal matrix of control signal equivalent Coulomb friction for x, y,

	and z axes
F_k	desired feedrate of the *k*th spline segment
$F_{\max}$	maximum feedrate (i.e. tangential velocity)
F_{tol}	feedrate search tolerance
$G^n : G^0, G^1, G^2$	geometric continuity: zero, first and second order
J_k	tangential jerk from *k-1*th to the *k*th spline segment
$J_{\max}$	maximum tangential jerk
$\mathbf{L}$	curve fitting constraint matrix
$\mathbf{L}^{(0)}$	position boundary constraint matrix
$\mathbf{L}^{(1)}$	first derivative boundary constraint matrix
$\mathbf{L}^{(2)}$	second derivative boundary constraint matrix
L_k	arc length of the *k*th segment
$\mathbf{M}$	diagonal matrix of control signal equivalent masses for x, y, and z axes
MSE	Mean Square Error (i.e. variance)
NURBS	Non Uniform Rational B-spline
N	number of spline segments
$N_{i,p}$	B-spline basis function - *i* is the knot span and *p* is the degree
$N_{i,p}^{(1)}$	first derivative B-spline basis function
$N_{i,p}^{(2)}$	second derivative B-spline basis function
N_t	number of time samples
N_w	number of spline segments in look-ahead window
N_Σ	number of integration points
$\mathbf{P}_i$	*i*th NURBS control point
$\mathbf{P}_{k,i}$	*i*th control point in the *k*th spline segment
$\mathbf{P}_x$	x-axis coordinates of all NURBS control points
$\mathbf{Q}_i$	*i*th data point
$\mathbf{Q}_x$	x-axis coordinates of all data points
SMC	Sliding Mode Control
S_k	*k*th spline segment
S_{ke}	arc displacement at the end of the *k*th spline segment
TCP	Tool Center Point
T_s	sampling period
$T_{a,k}$	time duration of the *k*th spline segment's constant acceleration phase
$T_{f,k}$	time duration of the *k*th spline segment's constant feedrate phase
$T_{j,k}$	time duration of the *k*th spline segment's constant jerk phase

$T_{\Sigma,k}$	total time duration of the *k*th spline segment
$\mathbf{U}=\{u_0,\ldots,u_m\}$	knot vector
a	start of interval on the spline parameter
$\mathbf{a}:\mathbf{a}_{\max},\mathbf{a}_{\min}$	axis acceleration vector: maximum and minimum limits
a_w	*w*th acceleration in look-ahead window calculation
$\hat{a}_k$	tangential acceleration at time sample *k* obtained with numerical differentiation
b	end of interval on the spline parameter
b_x	control signal equivalent viscous damping of the x-axis
d_{coulx}	control signal equivalent Coulomb friction of the x-axis
d_k	sum of chord lengths between data points in the *k*th spline segment
df	feed scanning resolution
$\mathbf{e}_x,\mathbf{e}_y$	axis tracking errors in x and y axes
f_{acc}	feed limit due to axis acceleration limits
f_{de}	feedrate at the end of the *d*th phase
f_{high}	upper bound feedrate
f_{jerk}	feed limit due to axis jerk limits
f_{low}	lower bound feedrate
f_{mid}	bisected feedrate
f_{vel}	feed limit due to axis velocity limits
f_w	*w*th feedrate in look-ahead window calculation
$\mathbf{j}:\mathbf{j}_{\max},\mathbf{j}_{\min}$	axis jerk vector: maximum and minimum limits
$\hat{j}_k$	tangential jerk at time sample *k* obtained with numerical differentiation
$l(a,b)$	approximated arc length on the interval [*a*,*b*] using Simpson's rule
m_x	control signal equivalent mass of the x-axis
n	number of NURBS control points minus one
n_k	number of NURBS control points minus one in the *k*th spline segment
$\mathbf{p}_{xk}$	x-axis coordinates of the *k*th spline segment control points
p	degree of NURBS curve
$\mathbf{q}_{k,i}$	*i*th data point in the *k*th spline segment
$\hat{\mathbf{q}}_{k,i}$	predicted *i*th data point in the *k*th spline segment
$\hat{\mathbf{q}}_{xk}$	predicted x-axis coordinates of the *k*th spline segment data points
s	arc displacement
$\dot{s}$	feedrate (i.e. tangential velocity)
$\ddot{s}$	tangential acceleration
$\dddot{s}$	tangential jerk

s_k	arc displacement along the *k*th spline segment
s_{de}	arc displacement at the end of the *d*th phase
$s_{p1}, s_{p2}, s_{p3}, ...$	arc displacement interval dividers
t	time
u	spline parameter
$\hat{\mathbf{u}}$	predicted spline parameter values
$\bar{u}_{k,i}$	spline parameter value assigned to data point $\mathbf{q}_{k,i}$
u_t	control signal (i.e. actuation torque)
$\mathbf{u}_t : \mathbf{u}_{t\max}, \mathbf{u}_{t\min}$	axis control signal vector: maximum and minimum limits
(u_i, s_i)	numerical integration point
$\hat{v}_k$	tangential velocity at time sample *k* obtained with numerical differentiation
w_i	weight of a NURBS control point
x	x-axis position
x_{meas}	measured position
x_{ref}	reference position
y	y-axis position
z	z-axis position
①,②,③,...	sub-segments

Chapter 1

Introduction

1.1 Introduction

With the growing demands to machine complex dies, moulds, aerospace, automotive, and biomedical parts in shorter cycle time, the utilization of Non Uniform Rational B-spline (NURBS) toolpaths has become more important than ever before. NURBS curves and surfaces have been accepted as standard modelling tools in computer-aided design (CAD) systems and have also been incorporated into computer-aided manufacturing (CAM) systems by industrial forerunners such as ESPRIT® and Siemens®. The advantages of NURBS toolpaths over the conventional linear and circular toolpath definitions are that they achieve faster feedrates, higher accuracy, and better surface finish in machining parts with complex geometry.

Computer Numerical Control (CNC) machine tools that utilize NURBS interpolation provide a significant competitive advantage to part manufacturers in terms of faster production rates and shorter time-to-market. The objective of this research is to develop new trajectory generation algorithms using NURBS toolpaths that reduce production cycle times in order to provide significant cost savings to part manufacturers. However, CNC machines are also a large investment, particularly for the small-to-medium sized enterprises. In order to make this technology accessible, it is important to make it modular, portable and low cost. Re-design of the CNC controller with new trajectory generation algorithms that incorporate parametric curve interpolation, smooth feedrate modulation, and feedrate optimization strategies is a practical approach to meet the high demands on part quality and fast production time while reducing manufacturing costs. Moreover, as NURBS becomes more widespread and mainstream, these algorithms can easily be implemented on low cost motion controllers. Inexpensive controllers will be able to deliver higher performance results. It is also desirable to integrate the new ideas with existing CNC's. One possibility could be to generate the optimized trajectories offline then download them to the controller's hard drive for real-time playback. Another is a semi-offline approach, wherein optimized feedrates for

S-curve profiles are determined offline, and then downloaded to the controller which is generally equipped to handle S-curve feed profiles for real-time calculation of the trajectory command positions.

There are several challenges associated with realizing a successful NURBS trajectory generator. First of all, the toolpath interpolator should minimize unwanted feedrate fluctuations while being numerically efficient and robust against accumulating round-off errors. Feedrate fluctuations are artefacts of arc-length parameterization errors, caused by an inaccurate mapping between the spline parameter and the arc displacement along the spline toolpath during interpolation. Discontinuity in the feed profile results in unsmooth tool motion, which causes visible feed marks on the machined part. Moreover, if the discontinuity produces high acceleration and jerk, then motor torque saturation and excitation of the machine tool's structural modes, which have the effect of degrading the positioning performance, may be encountered. When axis servo errors become excessive, the part geometry gets distorted and machining tolerances may be violated. For these reasons, it is important to implement the interpolator such that the feedrate can be accurately controlled. The feed modulation, on the other hand, needs to be able to continuously adjust the feedrate along different segments of the toolpath while ensuring that the final trajectory is jerk limited in all axes. Kinematic compatibility conditions between position, velocity, acceleration, and jerk should never be violated. The ability to perform feedrate modulation allows feedrate optimization strategies to be implemented, which ensure that high accuracy can be maintained throughout the toolpath without compromising the speed in low curvature segments. A look-ahead module is required in order to plan sufficient distance for accelerations and decelerations and to resolve kinematically infeasible cases. Additionally, in order to be practical for real-time implementation, the feed modulation must be computationally efficient.

Finally, reduction in the cycle time is due in large part to the feedrate optimization component. The minimum time feedrate optimization problem with jerk constraints is non-linear and non-convex, and consequently, not easily solvable. In general, obtaining an optimal feed profile requires a forward and backward traversal of the entire toolpath, which is impractical to perform in a real-time environment for very long toolpaths. To address this limitation, the toolpath can be divided into multiple segments and local near-optimal solutions can be computed. Unfortunately, there is no standard solution technique for this type of problem, but a customized solution methodology can be formulated given specific knowledge of the problem. Simplistic approaches tend to result in conservative feed profiles while gradient-based optimization techniques tend to be computationally expensive. A method that generates a feed profile with shorter cycle times than the simplistic solutions, with significantly less computational load compared to the gradient-based techniques, is sought.

In this thesis, a robust and numerically efficient NURBS interpolation strategy is developed, and contains integrated feedrate modulation and feedrate optimization functionalities. Unwanted feedrate fluctuations and sensitivity to round-off errors are avoided by applying the *feed correction polynomial* concept to NURBS toolpaths. A numerically efficient feedrate modulation strategy is developed, based on the trapezoidal acceleration profile, which guarantees that the final trajectory is limited in jerk in all axes. Furthermore, kinematic continuity is achieved by enforcing compatibility conditions between connecting segments throughout long toolpaths. The feed modulation strategy can be integrated with various feed optimization techniques as well. Specifically, a heuristic feedrate optimization method, that is computationally efficient, is developed and tested alongside the NURBS interpolation scheme. By utilizing insights into the physical constraints of the problem, the solution converges quickly and infeasible solutions are resolved in an efficient manner. Effectiveness of the overall NURBS trajectory generator is demonstrated in 3-axis machining experiments of a benchmark contour toolpath and a complex sculptured surface, which was derived from a biomedical implant.

Chapter 2

Literature Review

2.1 Introduction

Extensive work on motion planning and trajectory generation in both robotics and machining fields has been undertaken in pursuance of increased productivity and reduced costs for manufacturing processes. This chapter presents a review of literature and industrial state-of-the-art in the areas of NURBS toolpath generation, feedrate generation, and feedrate optimization. The tasks of toolpath generation are distributed over two systems - the CAD/CAM system and the CNC controller. The distribution of these tasks, as in this thesis, is illustrated in Figure 2-1. Computationally intensive tasks such as the toolpath parameterization and integration of the segment arc-length are generally handled by the CAM system in an offline environment, whereas feed generation and trajectory interpolation are realized in the CNC controller in real-time. Feedrate generation and optimization are interfaced subtasks of the trajectory generation module in the CNC controller. In the following, Section 2.2 gives a brief introduction to NURBS curve representation for toolpaths. A review of toolpath parameterization methods is presented in Section 2.3. Various spline interpolation techniques for parametric curves are surveyed in Section 2.4. Feed generation and optimization methods are explored in Sections 2.5 and 2.6, respectively. Conclusions for the chapter are presented in Section 2.7.

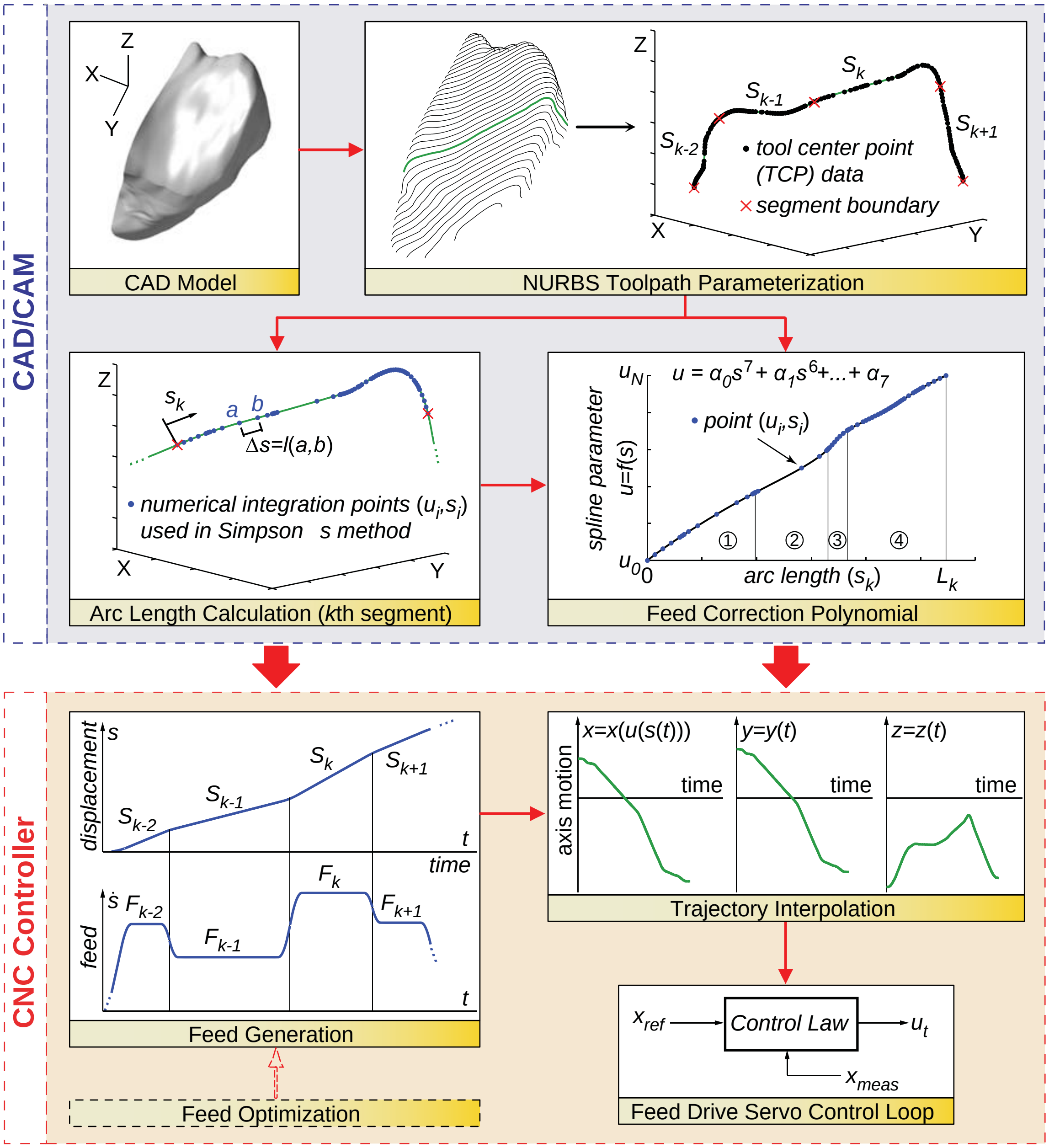

Figure 2-1. Overview of toolpath command generation as developed in this thesis. The CAD/CAM system handles toolpath parameterization, arc length calculation and feed correction polynomial fitting, while the CNC controller performs feed generation, feed optimization and trajectory interpolation.

2.2 Non Uniform Rational B-Splines (NURBS) Toolpaths

Conventionally, curved toolpaths are described with small linear and circular segments that are simple to interpolate. However, conventional methods are no longer sufficient to meet the growing demands on productivity and part quality. Research has shown that parametric spline interpolation has proven to be superior to linear and circular interpolation in terms of smoother and more continuous motion, which leads to better surface finish and faster feedrates [1] [2] [3] [4]. CAD models have long been able to utilize splines to design free-form contours and surfaces. However, only recently have splines started to become incorporated into industry standards and integrated into commercially available CAM systems and CNC controllers. Spline representation of curved toolpaths has two main advantages. First, the amount of data required to define spline segments is much less than that required to represent the same curve with linear and circular segments. Second, the continuity between segments allows for smoother motion that doesn't incur high jerk. Smoother motion improves the machine's positioning performance.

Non uniform rational B-spline (NURBS), which is a generalization of basis spline curves such as Bézier and nonrational B-splines, is favorable for toolpath generation because it offers a mathematically precise representation of freeform surfaces [5]. Most designers find them geometrically intuitive. Furthermore, NURBS curves and surfaces developed in the CAD model can be used for toolpath planning in the part program, which would mean no loss of accuracy in the post processing routines. Loss in accuracy naturally happens when lines and circles are used to approximate curves. However, even if the CAD model is not defined using NURBS, toolpath parameterization can still be performed on standard CAD output data comprising of small linear motion commands to realize the cycle time reduction with spline interpolation.

NURBS curves are defined by degree, control points, a knot vector, and weights. The degree of a cubic NURBS curve is three; for quintic, the degree is five. The order of a NURBS curve is the degree plus one. The order is also equal to the minimum number of control points that are required to define a p-degree NURBS curve. The number of control points is denoted by $n + 1$. The knot vector is a set of monotonically increasing values in the parametric space and it determines the realm of influence that each control point has on the NURBS curve. The knot vector divides the parameter space into intervals known as knot spans. As the spline parameter enters a new knot span, a new control point starts to exert its influence on the curve and an old control point no longer has effect. Calculating a point on the curve is done by taking a weighted sum of the control points where weighting factors are determined by evaluating the B-spline basis functions at the spline parameter and multiplying

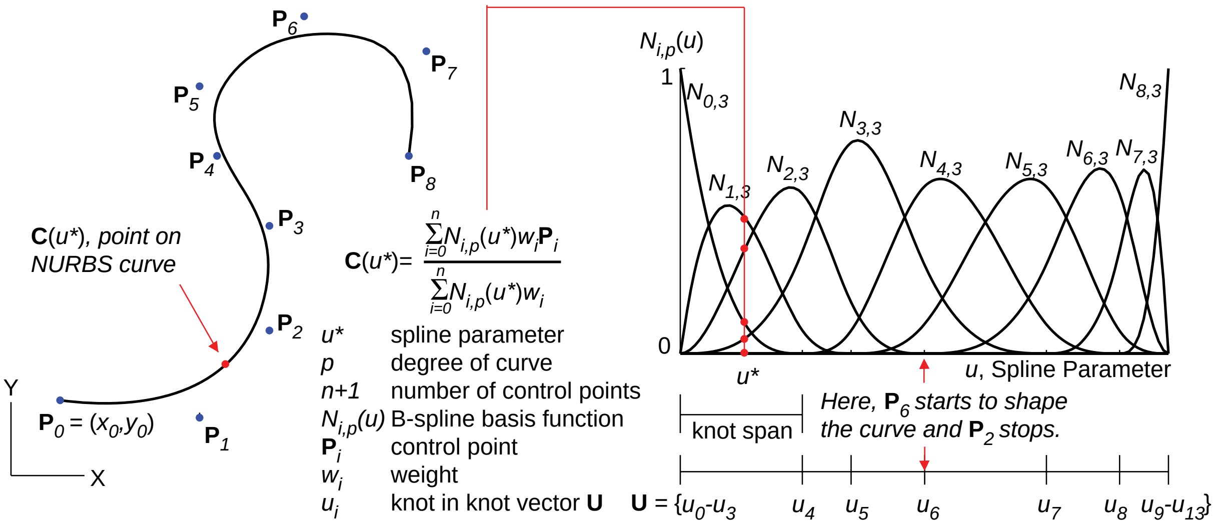
$\mathbf{P}_6$
$\mathbf{P}_7$
$\mathbf{P}_5$
$\mathbf{P}_4$
$\mathbf{P}_8$
$\mathbf{P}_3$
$\mathbf{C}(u^*)$, point on NURBS curve
$\mathbf{C}(u^*) = \frac{\sum_{i=0}^{n} N_{i,p}(u^*) w_i \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u^*) w_i}$
$\mathbf{P}_2$
Y
$\mathbf{P}_0 = (x_0, y_0)$
$\mathbf{P}_1$
X
u^* spline parameter
p degree of curve
$n+1$ number of control points
$N_{i,p}(u)$ B-spline basis function
$\mathbf{P}_i$ control point
w_i weight
u_i knot in knot vector $\mathbf{U}$
$N_{i,p}(u)$
1
0
$N_{0,3}$
$N_{1,3}$
$N_{2,3}$
$N_{3,3}$
$N_{4,3}$
$N_{5,3}$
$N_{6,3}$
$N_{7,3}$
$N_{8,3}$
u^*
u, Spline Parameter
knot span
Here, $\mathbf{P}_6$ starts to shape the curve and $\mathbf{P}_2$ stops.
$\mathbf{U} = \{u_0\text{-}u_3 \quad u_4 \quad u_5 \quad u_6 \quad u_7 \quad u_8 \quad u_9\text{-}u_{13}\}$

Figure 2-2. NURBS curve representation. [5]

them by the weights assigned to individual control points. An example of how a 2-D NURBS curve (p = 3; n = 8) is constructed is illustrated in Figure 2-2.

Control points are the geometric parameters that define the shape of the curve. A change in a control point's position results in a visible direct effect in shaping the curve locally. Weights also characterize the extent of a control point's influence on the curve's shape. Increasing the weight of a control point pulls the points on the curve affected by that control point closer to it. When a weight approaches infinity the curve will pass through the corresponding control point. On the other hand, decreasing the weight pushes the curve away from the corresponding control point, where a weight of zero eliminates all influence. The shape modification effects of control points and weights are illustrated in Figure 2-3.

2.3 Toolpath Parameterization

Toolpath parameterization is the task of obtaining a mathematical representation of a toolpath such that the position coordinates of the tool tip can be computed in terms of an independent variable called the spline parameter. This task takes place in the CAD/CAM system as shown in Figure 2-1. The most important requirements of the toolpath parameterization module are to generate splines that are geometrically continuous and to accurately describe the machining geometry. Spline segments that have common boundary positions are said to have G^0 continuity. If the unit tangent vectors at the segment boundary

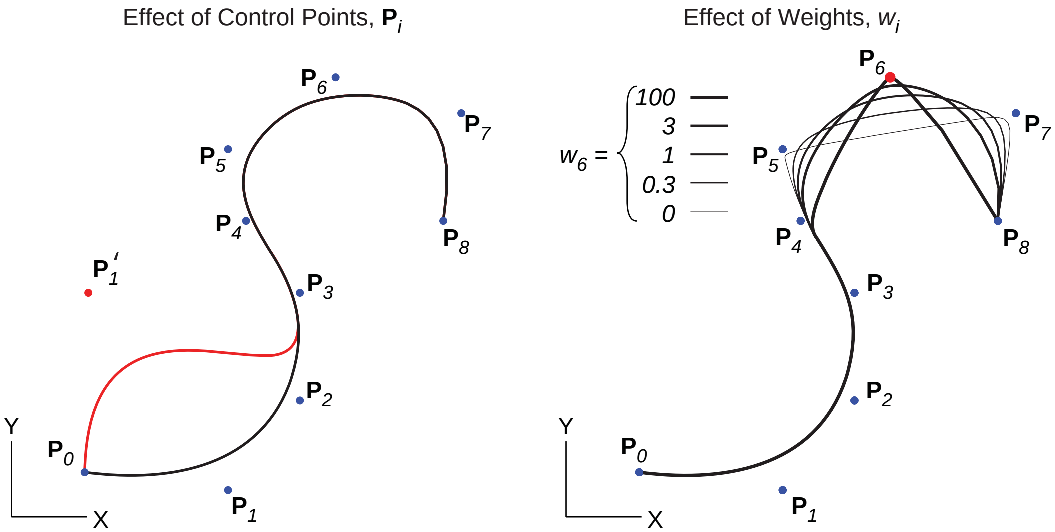

Figure 2-3. Shape modification of a NURBS curve using control points and weights. [5]

are equal, then the splines are G^1 continuous. Segments are G^2 continuous if they share a common center of curvature at the boundary. At the least, G^2 continuity is required to achieve smooth motion in CNC machining. Parametric derivative continuity, denoted as C^n continuity where n is the order of the derivative, is a special case of geometric continuity when the parameterization is with respect to the distance traveled along the toolpath, which is also known as arc-length parameterization. In the case of arc-length parameterization for toolpaths, C^2 continuity is necessary, however when the toolpath is not arc-length parameterized, G^2 continuity is sufficient and is much more flexible than parametric continuity constraints [6] [7].

Formulating a NURBS curve involves obtaining a knot vector, weights, and control points, whereas for power basis polynomial representations, the only unknowns are the algebraic coefficients. Cubic and quintic polynomial splines have been heavily investigated. Wang and Yang [8] developed a spline curve fitting algorithm that starts by obtaining a chordal-length parameterized cubic spline; then based on the cubic spline finds a nearly arc-length parameterized quintic spline (NAPQS) with C^2 continuity. Wang et al. [9] furthered this technique to produce approximately arc-length parameterized C^3 quintic interpolatory splines (AAP C^3 QIS) by utilizing an iteration to optimize the parameterization and adding a third derivative continuity condition. Later, Erkorkmaz and Altintas [10] formulated and solved an unconstrained minimization problem with an analytically integrable objective function to yield optimally arc-length parameterized (OAP) quintic splines. On the other hand, NURBS curve and surface fitting has been described in detail by Piegl and Tiller [5], whose methodology has been the basis for most NURBS parameterization techniques.

In implementing the basic NURBS parameterization methods described in literature, the main issue was achieving curvature (i.e. second derivative) continuity at segment connections without introducing oscillations into the toolpath. The oscillations are undesirable because they cause the toolpath to deviate from the desired geometry and also cause unsmooth motion. To address this issue, Lee and Liang [11] modified the least squares curve fitting objective function to include a *strain energy* minimizing term, based on the integral of the squared curvature. Their rationale was to penalize high curvature, thereby reducing the oscillatory behaviour in the curve. For the same purpose, Sencer [12] presented a smoothening term which penalized high jerk values. In this thesis, *beta-constraints*, which are mathematical relationships that determine if two parametric curves connect with geometric derivative continuity [6] [7], are investigated.

Barsky and DeRose introduced beta-constraints for applications in computer graphics to test the smoothness of two connected parametric curves, which may have different parameterizations. Their definition of geometric continuity is as follows:

"**Definition 1**. Let $\mathbf{q}(u)$ and $\mathbf{r}(t)$ be two regular C^n parameterizations meeting at a point $\mathbf{J}$. They meet with *n*th-order geometric continuity, denoted G^n, if there exists a parameterization $\widetilde{\mathbf{q}}$ equivalent to $\mathbf{q}$ such that $\widetilde{\mathbf{q}}$ and $\mathbf{r}$ meet with C^n continuity at the point $\mathbf{J}$." [6]

In other words, if it is possible to reparameterize one of the curves such that the two curves meet with C^2 parametric continuity, then the two curves join smoothly. For example, consider two parametric curves, $\mathbf{q}(u)$, $u \in [0,1]$, and $\mathbf{r}(t)$, $t \in [0,1]$ that meet with G^2 continuity at the junction $\mathbf{r}(0) = \mathbf{q}(1)$. Then there exists a scalar function, $u(\widetilde{u})$, that maps $\widetilde{u} \in [\widetilde{u}_0, 1] \Rightarrow u \in [0,1]$ without changing the shape of $\mathbf{q}(u)$, such that $\mathbf{q}(u(\widetilde{u})) = \widetilde{\mathbf{q}}(\widetilde{u})$ [6]. From Definition 1:

$$\begin{aligned}\mathbf{r}'(0) &= \tilde{\mathbf{q}}'(1)\\ \mathbf{r}''(0) &= \tilde{\mathbf{q}}''(1)\end{aligned} \tag{2.1}$$

Using the chain rule, the first and second derivatives of $\tilde{\mathbf{q}}(\tilde{u}) = \mathbf{q}(u(\tilde{u}))$ can be written as:

$$\begin{aligned}\tilde{\mathbf{q}}'(\tilde{u}) &= \mathbf{q}'(u(\tilde{u}))\cdot u'(\tilde{u})\\ \tilde{\mathbf{q}}''(\tilde{u}) &= \mathbf{q}''(u(\tilde{u}))\cdot u'(\tilde{u})^2 + \mathbf{q}'(u(\tilde{u}))\cdot u''(\tilde{u})\end{aligned} \tag{2.2}$$

Given that $u(\tilde{u}=1)=1$ from the mapping, the first and second derivatives of $\tilde{\mathbf{q}}$ evaluated at $\tilde{u}=1$ are:

$$\begin{aligned}\tilde{\mathbf{q}}'(1) &= \mathbf{q}'(1)\cdot u'(1)\\ \tilde{\mathbf{q}}''(1) &= \mathbf{q}''(1)\cdot u'(1)^2 + \mathbf{q}'(1)\cdot u''(1)\end{aligned} \tag{2.3}$$

In [6], $u'(1)$ and $u''(1)$ are substituted with beta values $\beta 1$ and $\beta 2$, where $\beta 1$ must be greater than zero to preserve the direction of the tangent vector, and $\beta 2$ can be any real value. Substituting in the beta values and Equation (2.3) into Equation (2.1) yields the beta-constraints for first and second order geometric continuity.

$$\begin{aligned}\mathbf{r}'(0) &= \beta 1\cdot \mathbf{q}'(1)\\ \mathbf{r}''(0) &= \beta 1^2\cdot \mathbf{q}''(1) + \beta 2\cdot \mathbf{q}'(1)\end{aligned} \tag{2.4}$$

Equation (2.4) is used in the NURBS toolpath parameterization to generate segmented toolpaths, which have G^2 continuity. Rather than enforcing parametric continuity, geometric derivative continuity constraints are imposed because they allow for further shaping of the NURBS toolpath in order to eliminate oscillations, while ensuring continuity at segment boundaries. In this thesis, Chapter 3 presents a NURBS toolpath parameterization that utilizes beta-constraints to guarantee G^2 continuous toolpaths.

2.4 Spline Interpolation

In the CNC controller, trajectory interpolation of the spline toolpath is performed to obtain a commanded position at each sample time step as shown in the "Trajectory Interpolation" block in Figure 2-1. In this stage, the spline parameter is transformed into the time domain and the toolpath is converted into a machining trajectory which takes into account machining conditions such as feed, acceleration, actuating torques, and jerk [8]. For an arc-length parameterized spline $C(s)$, where s is the arc-length, conversion to the time domain only requires the calculation of arc-length positions $s(t)$, based on a feed generation technique at each time step, $t = kT_s$, where T_s is the sampling period and k is an integer value between zero and the total number of time steps, N_t. Substituting those values directly into the parametric curve equation yields position commands at each time step, i.e. $C(s(t)) = C(t)$. Toolpaths that are not parameterized according to their arc-length require an additional transformation from the spline parametric space to the arc-length displacement along the curve. Arc-length positions at each time step are converted to spline parameter values with the mapping defined by $u(s)$ and substituted into the parametric definition of the curve such that $C(u) \to C(u(s(t))) = C(t)$. The challenge in implementing a spline interpolator is efficiently calculating the spline parameter accurately to achieve the desired arc displacement increment along the toolpath at each time step. If the tool tip does not travel the specified arc displacement, then feedrate fluctuations result and lend to high acceleration and jerk values, which are detrimental to the part quality and the machine tool's life. Erkorkmaz and Altintas [10] reported feed fluctuations on the order of 0.2% for the 88-segment fan-shaped toolpath used by Wang et al. [8] [9], and up to 78% feed fluctuation for a spline toolpath composed of ten random points. In the former case, although the feed fluctuations may not seem high, the resulting oscillations it causes in the acceleration and jerk profiles are still more significant and tie into the part quality and machine life. It is also important to note that the mapping between the spline parameter and the arc-length needs to be robust against numerical round-off and accumulation errors.

Natural interpolation assumes a proportional relationship between the spline parameter u, and the arc-length s, as shown in Equation (2.5). Here, L is defined as the segment's arc-length, T_s is the sampling period and N_t is the total number of time steps. The domain of the spline parameter, u, is assumed to be between zero and one.

$$u(t) = s(t)/L \qquad \textit{where} \quad \begin{array}{l} t = kT_s \\ k = 0,1,\ldots,N_t \end{array} \tag{2.5}$$

Basically the curve is being interpolated at constant spline parameter increments. This method is generally sufficient for approximately arc-length parameterized toolpaths. However, when this relationship does not hold, better estimates of the spline parameter can be computed with a Taylor series approximation as follows:

$$u_k = u_{k-1} + \underbrace{\dot{u}_k T_s}_{\substack{1st\\ order\\ term}} + \underbrace{\frac{1}{2}\ddot{u}_k T_s^2}_{\substack{2nd\\ order\\ term}} + \underbrace{H.O.T.}_{\substack{higher\\ order\\ terms}} \qquad where \quad k = 0,1,\ldots,N_t \tag{2.6}$$

Above, u_k and u_{k-1} are the spline parameter values at the current and previous time steps, respectively; $\dot{u}_k$ and $\ddot{u}_k$ are the first and second derivatives of the spline parameter with respect to time, calculated at the current time step. Huang and Yang [13] presented a first-order Taylor series approximation to realize the desired feedrate, which has been successfully implemented in commercially available CNC systems [14]. Lin [15] proposed a second-order approximation that was reported to achieve better accuracy, naturally at higher computational cost, which further reduced the magnitude of the feedrate fluctuations. The downfall of Taylor series approximations is the inevitable accumulation of numerical errors due to the recursive addition and rounding. To eliminate these errors, Erkorkmaz and Altintas [16] developed an iterative approach to solve for the spline parameter utilizing a high-order polynomial relationship between the desired arc increment and parameter increment. Despite the high-order, in general less than three iterations were required to converge on a solution. Another technique proposed by Cheng et al. [17] is a predictor-corrector algorithm for better feedrate control. Furthermore, Erkorkmaz and Altintas [10] introduced the use of a feed correction polynomial. For example,

$$u = \alpha_0 s^7 + \alpha_1 s^6 + \alpha_2 s^5 + \cdots + a_6 s + a_7 \tag{2.7}$$

Utilizing numerical evaluations of the arc-length at incremental spline parameter values, a 7th order polynomial is obtained to express the spline parameter in terms of the derived arc-length as shown in the "Feed Correction Polynomial" block in Figure 2-1. Lei et al. [18] developed a similar concept with cubic Hermite splines and called it the inverse length function (ILF). In this thesis, feed correction is chosen to be investigated for NURBS interpolation because of its robustness against accumulating round-off errors, which is prone to happen with a truncated Taylor series expansion. Also, the polynomial can be evaluated more efficiently compared to iterative techniques, thereby leaving more resources for feedrate optimization and other CNC functions.

The main challenge encountered when fitting a feed correction polynomial for NURBS segments is capturing the relationship between the spline parameter and arc-length with a single polynomial, when the toolpath geometry is significantly complex. In this thesis, a method for fitting multiple 7th-order polynomial splines is developed based on the complexity of the toolpath and a pre-specified mean square error (MSE) tolerance on the fitting error. Chapter 3 explains the feed correction polynomial as applied to NURBS toolpaths.

2.5 Feed Generation

Feed generation takes place in the CNC controller as shown in Figure 2-1 and characterizes the motion along the toolpath in terms of the arc displacement $s(t)$, feed $\dot{s}(t)$, acceleration $\ddot{s}(t)$, and jerk $\dddot{s}(t)$ in the tangential direction. To achieve smooth motion, the displacement, feedrate and acceleration profiles must be continuous throughout the toolpath. The feed generation must also limit the jerk, in order to achieve high performance tracking and avoid exciting the structural modes of the machine tool. Several feed profiles have been suggested in the literature. Erkorkmaz and Altintas [16] presented a jerk-limited feed profile composed of piecewise constant jerk values. Acceleration transients demonstrate a characteristic trapezoidal profile and the feedrate exhibits an S-curve profile which has parabolic transitions. The formulation is simple and the computational load is small. Macfarlane and Croft [19] proposed a jerk-bounded trajectory that employed an approximation of a sine wave for the acceleration ramps. A jerk-continuous profile was also introduced by Erkorkmaz [20] that utilized cubic acceleration transients which are obtained by integrating quadratic jerk functions. In the same vein, Pritschow integrated a squared sine jerk function, $(\sin^2)$ [21]. The last three methods are favorable for better continuity. These feed generation techniques are applied to a single toolpath segment and are scalable to longer toolpaths, by considering the kinematic compatibility conditions between adjacent segments.

Recently, methodologies that generate multiple segment profiles have started to be reported in the literature. Lin et al. [22] proposed a method for scheduling S-shape feed profiles with triangular acceleration/deceleration transients for multi-segmented toolpaths. Independently from their research, the work in this thesis follows a similar approach but differs in conceptualizing the implementation of the feed profiling.

There are two challenges related to successful feed generation. The first is implementing the feed profiling with the desired feedrate and specified limits on the acceleration and jerk such that kinematic compatibility is maintained. Kinematic compatibility is achieved if there is enough travel distance to perform the desired feed motion. The second is modulating the feed continuously for long toolpaths that have variable command feedrates for each segment. To address the first challenge, kinematic compatibility conditions must be checked to ensure that the desired motion can be physically carried out. For example, if a toolpath segment lacks the travel length to accelerate to the desired feedrate and subsequently decelerate to the specified end feedrate, then the desired feedrate must be modified to reflect an achievable feed transition within the specified acceleration and jerk bounds. Erkorkmaz and Altintas [16] derived four conditions on the jerk, acceleration, deceleration, and travel length for their proposed jerk-limited feed profile. In this thesis, similar compatibility conditions are derived for the proposed multi-segment framework. As for the second challenge, feed modulation has generally been handled by a look-ahead module that adjusts the feedrate at high curvature sections of the toolpath. According to the prescribed chord error tolerance, the feed is decreased as necessary and a re-interpolation of the feed profile is performed to generate acceleration and deceleration ramps that are jerk-limited considering the machine's dynamics [23] [24]. This method is as also known as two-stage interpolation [25]. Lin et al. [22] also consider the errors due to the servo control loop dynamics. However, there is a lack of direction on what to do when the designed feed profile runs into compatibility issues. For example, how to resolve the issue when the command feedrates cannot be realized smoothly, especially when toolpaths have multiple segments. In this work, the proposed framework performs feed modulation segment-by-segment concurrently with kinematic compatibility checks prior to any interpolation, which eliminates the second interpolation step and guarantees that the interpolated trajectory is smooth and continuous. Chapter 4 proposes a generalized framework for long toolpaths based on the jerk limited S-curve function, which is widely utilized in existing CNC`s.

2.6 Feed Optimization

In machining complex dies, molds, aerospace and automotive parts, or biomedical components, it is crucial to minimize the cycle time while preserving the quality and tolerance integrity of the part being produced. Optimization of the feed profile in machining NURBS toolpaths, which are becoming more widespread for producing freeform parts [26] [27] plays a major role in achieving this objective. Unfortunately, the feed optimization problem does not lend itself to a straightforward solution, especially when jerk limits in the individual axes need to be considered, in order to limit the amount of vibration and contouring error induced during rapid tool movements. Hence, extensive research has been dedicated to solving this problem.

A two-pass algorithm for minimum time trajectory planning of a robotic manipulator was developed in the seminal paper by Bobrow et al. [28], which yields the optimal feed profile subject to torque constraints. However, the resulting motion is jerky due to discontinuous actuator torques. Constantinescu and Croft [29] addressed this issue by also limiting the first derivative of actuator torques or the "torque rate," which produced smooth time-optimal trajectories. In the machining literature, Bobrow's technique has recently been extended to also incorporate jerk limits by Dong et al. [30]. Although theoretically successful, the requirement to perform full forward and backward passes limits the practicality of this approach, particularly for long toolpaths. A practical and highly effective look-ahead technique was proposed by Weck et al. [31], which is based on setting the feed limit for each segment by considering the local worst-case curvature. This method has a straightforward formulation and requires minimal computation, allowing convenient implementation in CNC interpolators. However, the cycle time reduction is mildly conservative, since the toolpath sections where higher feedrates are feasible are not fully utilized. An alternative strategy was proposed by Altintas and Erkorkmaz [32] that performs a gradient-based search among possible minimized-jerk feed profiles, which leads to shorter cycle times, but at the expense of significantly higher computational load. The latter approach may be more suitable for semi-offline process planning.

In general, the constraints for the feedrate optimization problem are based on the physical capabilities of the machine and other factors that affect the final part quality and surface finish. Cheng and Chin [33] investigated the causes of machining contour errors and in particular developed a system model that incorporated errors due to the cutting process, trajectory tracking, and the machine structure. The focus of this thesis is on the trajectory generation process in which the feedrate, acceleration and jerk are significant factors that affect the tracking performance. The feedrate is correlated to the cutting forces between the tool and the workpiece. Erdim et al. [36] developed a feedrate maximizing strategy that utilized a force-based model of the system cutting dynamics. As excessive cutting forces degrade the part quality, the feedrate was maximized subject to a maximum allowable cutting force. In this thesis, a maximum feed limit is incorporated to allow regulation of the cutting forces. Additionally, the physical limits of the actuators' speed and torque must also be respected. Actuator limitations can be expressed through constraints on the velocity and acceleration, as shown in [28]. It is also possible to replace the acceleration constraints with limits on the torque demand, which can be predicted by a dynamic model of the system, as was done in [32]. Similarly, Butler et al. [35] presented a feedrate generation method that yielded minimum travel time without actuator saturations, based on knowledge of the axis dynamics. Avoiding actuator saturation is necessary. If the actuators are saturated, then the system becomes non-linear, which can lead to instability. Lastly, it has been experimentally verified that the jerk of the desired trajectory can adversely affect the tracking control performance of robotic manipulators [34] and machine tools [16]. Limits on the jerk are necessary to achieve smooth motion. Thus in this thesis, feedrate optimization is performed subject to constraints on the feedrate, as well as, on the velocity, torque demand, and jerk in each axis.

Chapter 5 presents a new heuristic technique which yields shorter cycle time compared to the "worst-case" approach presented in [31], and converges to a feasible solution faster than gradient-based methods [32], within a deterministic number of iterations. The essence of the new technique is presented, along with benchmark experiments comparing the heuristic method to other approaches proposed in CNC literature. The feed optimization technique developed in this thesis is also in the process of being published in [37].

2.7 Conclusions

This chapter has presented a survey of academic literature and industrial practice relevant to NURBS toolpath planning, feedrate generation and optimization. To realize smooth and continuous motion, the spline interpolator must realize the desired arc displacement required by the commanded feed profile. Additionally, the feed generation method must ensure that the feed profile demonstrates acceleration continuity throughout the toolpath, while providing the capability to modulate the feed as necessary. To exploit the full potential of NURBS toolpaths, a feedrate optimization method is required to generate time-optimal trajectories subject to the dynamic constraints determined by the machine tool's physical and control capabilities. All three components together make up a command trajectory generator for a state-of-the-art CNC controller that meets the demands of high productivity and high quality, without incurring large capital costs. Hence, there is a strong need to develop NURBS trajectory generation algorithms to implement on existing and new low-cost machine tools, in a practical and reliable manner. In the following, Chapter 3 presents NURBS toolpath parameterization and interpolation methods. Chapter 4 describes the proposed feed generation framework for multi-segment toolpaths. Finally, Chapter 5 develops a feedrate optimization method that can be successfully integrated into the CNC controller with the aforementioned NURBS interpolation and feed generation components.

Chapter 3

NURBS Trajectory Generation

3.1 Introduction

In this chapter, a numerically robust and computationally efficient method for NURBS trajectory generation is presented. In Section 3.2, a parameterization method is designed which fits smooth and geometrically continuous NURBS curves to designated data points. Interpolation of the NURBS curve is performed with a feed correction polynomial that maps the distance traveled along the spline to the spline parameter. To obtain the feed correction polynomial, a constrained optimization problem is constructed and solved in Section 3.3, using the Lagrange Multipliers (LM) technique. Simulation results demonstrating the effectiveness of the developed trajectory interpolation method are presented in Section 3.4. The conclusions are presented in Section 3.5.

3.2 NURBS Toolpath Parameterization

Non uniform rational B-splines have been incorporated into state-of-the-art CAD/CAM software packages such as Unigraphics NX3 and CATIA V5. However, the use of NURBS in geometric modelling is much older than the use of NURBS in toolpath planning, which is still at an early stage. Hence, much work still remains to create advanced algorithms for NURBS to be practically used in trajectory generation. In order to test the trajectory generation methods in this thesis, the NURBS toolpaths need to be segmented and geometrically continuous up to the second derivative, which includes position, tangent and curvature continuity. Using a CAD/CAM package such as Unigraphics, it was found that only position continuity was maintained between segments in the toolpath generation. The discontinuities in the derivative profiles pose several problems for the CNC controller in terms of tracking performance. This served as the motivation for developing a curve fitting algorithm that produces segmented, curvature continuous NURBS toolpaths.

The objective of the NURBS curve fitting algorithm is to optimally place the control points, given assigned knots and weights, such that the error between the specified data

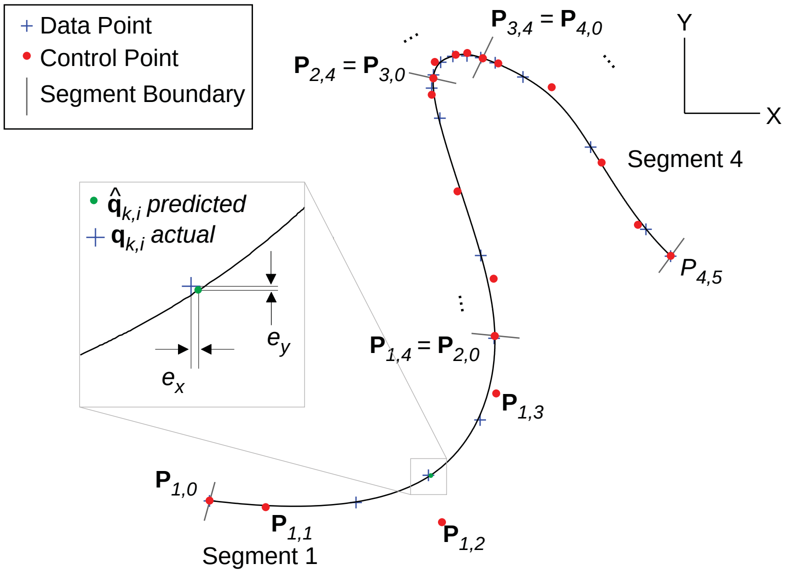
+ Data Point
● Control Point
│ Segment Boundary
● $\hat{\mathbf{q}}_{k,i}$ predicted
+ $\mathbf{q}_{k,i}$ actual
e_y
e_x
$\mathbf{P}_{3,4} = \mathbf{P}_{4,0}$
$\mathbf{P}_{2,4} = \mathbf{P}_{3,0}$
Y
X
Segment 4
$P_{4,5}$
$\mathbf{P}_{1,4} = \mathbf{P}_{2,0}$
$\mathbf{P}_{1,3}$
$\mathbf{P}_{1,0}$
$\mathbf{P}_{1,1}$
$\mathbf{P}_{1,2}$
Segment 1

Figure 3-1. Optimal placement of control points to fit multiple NURBS segments to data points.

points and the curve is minimized, as shown in Figure 3-1. At the same time, maintaining geometric continuity between the segments is desired to ensure that the toolpath is smooth. Geometric continuity constraints for position (G^0), tangent (G^1) and curvature (G^2) are imposed to guarantee smoothness at each segment boundary. Beta-constraints, which were developed by Barsky and DeRose [6] [7], are utilized here to impose these boundary conditions. A constrained optimization problem that minimizes the errors, e_x and e_y, between the specified data points and the segmented curve, while adhering to geometric constraints, is constructed. The control points are solved for using the Lagrange Multipliers technique.

As illustrated in Figure 2-2, a NURBS curve is represented by a knot vector, **U**, a set of control points, $\mathbf{P}_i$, and weights for each control point, w_i. The degree of the curve is denoted as p and the number of control points is n + 1. Defining the knot vector as shown in Equation (3.1), B-spline basis functions, $N_{i,p}$, are evaluated recursively at the spline parameter, u, with Equation (3.2).

$$\mathbf{U} = \{u_0, u_1, \ldots, u_{m=n+p}\} = \left\{ \underbrace{0,\ldots,0}_{p+1}, u_{p+1}, \ldots, u_{n-1}, \underbrace{1,\ldots,1}_{p+1} \right\} \tag{3.1}$$

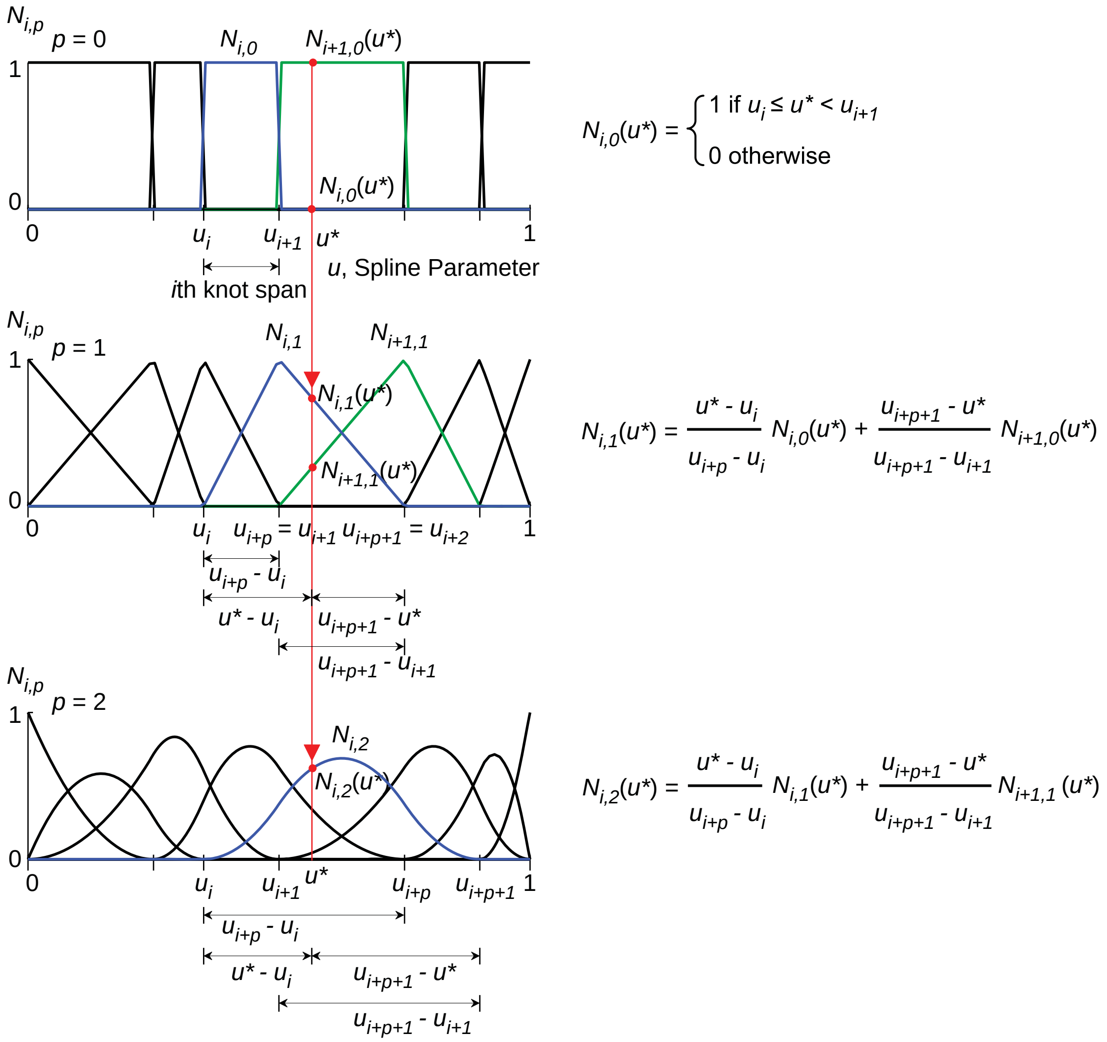

Figure 3-2. Evaluation of zero (*p*=0), first (*p*=1) and second (*p*=2) degree B-spline basis functions.

$$N_{i,0}(u) = \begin{cases} 1 & if \quad u_i \le u < u_{i+1} \\ 0 & \quad otherwise \end{cases}$$
$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1.p-1}(u) \tag{3.2}$$

The knots on the spline parameter axis are denoted by u_i and the *i*th knot span is defined as the spline parameter range between the *i*th and *i+1*th knot. Evaluation of the zero (*p*=0), first (*p*=1), and second (*p*=2) degree B-spline functions at u^* is demonstrated in Figure 3-2. u^* lies within the *i+1*th knot span. Thus, $N_{i,0}(u^*)$ is zero since u^* is outside the *i*th knot span, however $N_{i+1,0}(u^*)$ is one. First degree B-spline functions are simply a blend of the zero degree B-spline functions. $N_{i,1}(u^*)$ is a linear combination of the zero degree B-spline functions, $N_{i,0}$ and $N_{i+1,0}$ evaluated at u^*. The ratio of the *i*th span to the distance between u^* and the *i*th knot is the contribution of the $N_{i,0}(u^*)$ term, while the ratio of the *i+1*th span to the distance between u^* and the *i+2*th knot is the contribution of the $N_{i,1}(u^*)$ term. Similarly, second degree B-spline functions are blends of the first degree functions.

Combining the B-spline basis functions with the weights into a single term, the points on a NURBS curve are linear combinations of the control points, as shown in Equation (3.3). This property allows the construction of a linear system of equations for the constrained optimization problem used in determining the control point locations.

$$\mathbf{C}(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} = \sum_{i=0}^{n} R_{i,p}(u)\mathbf{P}_i, \qquad 0 \le u \le 1 \tag{3.3}$$

It is important to note that a *p*-degree NURBS curve must have at least *p* + 1 control points, which gives the requirement that $n \ge p$ since *n* + 1 is the number of control points. In order to fit multiple NURBS segments to the designated data points, we must first arrange the number of data points and the number of control points for each curve segment such that *n* is greater than or equal to *p*. The number of control points is set to be equal to the number of data points so that only one value needs to be selected, while still maintaining full rank for the optimization problem. Given a set of data points $\mathbf{Q}_i = [x_i \quad y_i \quad z_i]^T$ of size *M* + 1, and the desired degree *p* and value for *n*, the data points are segment such that each NURBS segment has *n* + 1 data points, where the last data point of a segment is also the first data point of the next adjacent segment, as shown in Figure 3-3. For example, if *p* = 2 and *n* = 3, each data point $\mathbf{Q}_i$ is assigned to a placement holder, $\mathbf{q}_{k,i}$, which represents the *i*th data point in the *k*th segment, as shown in Table 3-1.

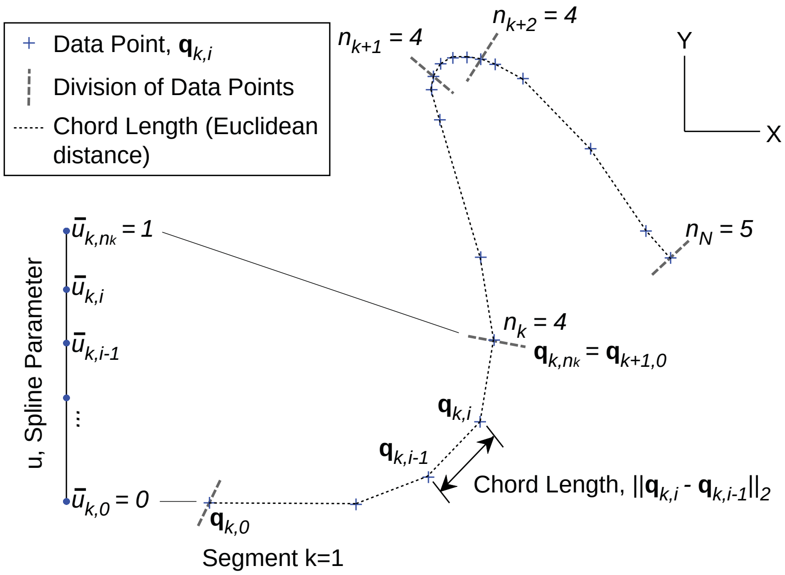

Figure 3-3. Division of data points and computation of spline parameter values for each data point (p = 3, n = 4).

In some cases, the number of data points in the last segment will not be sufficient to fit a p-degree NURBS curve. To resolve this issue, the data points are absorbed into the second last segment. An example of this circumstance is illustrated in Figure 3-3, where the number of data points is 18, p = 3 and n = 4. The last segment is assigned six control points, hence, $n_N = 5$, where N is the total number of segments.

To construct the curve fitting optimization problem, a system of linear equations is composed of predictions that correspond to the data points. First, each data point is assigned a spline parameter value, $\bar{u}$, based on the chord length parameterization method described by

Table 3-1. Segmentation of data points.

Segment, k	$i = 0$	$i = 1$	$i = 2$	$i = 3 = \mathrm{n}$
k=1	$\mathbf{q}_{1,0} = \mathbf{Q}_0$	$\mathbf{q}_{1,1} = \mathbf{Q}_1$	$\mathbf{q}_{1,2} = \mathbf{Q}_2$	$\mathbf{q}_{1,3} = \mathbf{Q}_3$
k=2	$\mathbf{q}_{2,0} = \mathbf{Q}_3$	$\mathbf{q}_{2,1} = \mathbf{Q}_4$	$\mathbf{q}_{2,2} = \mathbf{Q}_5$	$\mathbf{q}_{2,3} = \mathbf{Q}_6$
k=3	$\mathbf{q}_{3,0} = \mathbf{Q}_6$	$\mathbf{q}_{3,1} = \mathbf{Q}_7$	$\mathbf{q}_{3,2} = \mathbf{Q}_8$	$\mathbf{q}_{3,3} = \mathbf{Q}_9$
k=4	$\mathbf{q}_{4,0} = \mathbf{Q}_9$	$\mathbf{q}_{4,1} = \mathbf{Q}_{10}$	$\mathbf{q}_{4,2} = \mathbf{Q}_{11}$	...

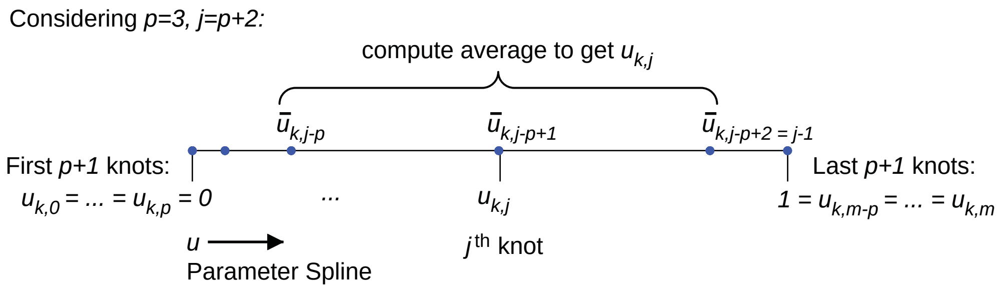

Figure 3-4. Sample calculation of the jth knot in a knot vector (*p* = 3, *n* = 5).

Piegl and Tiller [5], as illustrated in Figure 3-3. Let $\bar{u}_{k,i}$ represent the spline parameter for the *i*th data point in the *k*th segment. Here, $\| \cdot \|_2$ denotes the Euclidean distance between two data points (i.e. chord length), and d_k is the sum of the chord lengths in the *k*th segment. Also, $n_k + 1$ is the number of control points in the *k*th segment.

$$d_k = \sum_{i=1}^{n_k} \| \mathbf{q}_{k,i} - \mathbf{q}_{k,i-1} \|_2$$

$$\bar{u}_{k,0} = 0 \qquad \bar{u}_{k,n_k} = 1 \tag{3.4}$$

$$\bar{u}_{k,i} = \bar{u}_{k,i-1} + \frac{\| \mathbf{q}_{k,i} - \mathbf{q}_{k,i-1} \|_2}{d_k}, \qquad i = 1, \ldots, n_k - 1$$

Then using the $\bar{u}$ values in the chord length parameterization, knot vectors, $\mathbf{U}_k$, are constructed for each segment based on an averaging method which is also described by Piegl and Tiller [5], which reflects the distribution of data points in the segment. The first and last $p + 1$ knots are assigned zero and one, respectively. There are $m + 1$ knots in total, where $m = n + p$. The interior knots are obtained by averaging $\bar{u}$ values with the formula in Equation (3.5). An example of calculating the *j*th knot is illustrated in Figure 3-4.

$$u_{k,0} = \cdots = u_{k,p} = 0 \quad u_{k,m-p} = \cdots = u_{k,m} = 1$$

$$u_{k,j} = \frac{1}{p} \sum_{i=j-p}^{j-1} \bar{u}_{k,i}, \qquad j = p+1, \ldots, n_k - 1 \tag{3.5}$$

Additionally, weights can be assigned to the unknown control points. For simplicity all weights are set to one, which reduces Equation (3.3) to a nonrational B-spline expression as follows:

$$\mathbf{C}(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i \tag{3.6}$$

The final step is to determine the optimal locations of the control points such that the Euclidean distances between the data points and the sample points on the NURBS curve are minimized, subject to boundary conditions between segments. Using the computed knot vectors, $\mathbf{U}_k$, B-spline basis functions are evaluated at each spline parameter, $\bar{u}_{k,i}$, and substituted into Equation (3.6) to generate curve point predictions that correspond to the data points. Since x, y, and z coordinates are independent of each other, a system of linear equations is created for each coordinate axis. Then, minimizing the errors in each axis minimizes the Euclidean distance between the predicted, $\hat{\mathbf{q}}_{k,i}$, and actual, $\mathbf{q}_{k,i}$, data points. For illustration, the constrained optimization problem is formulated for the x-axis. Similar formulations can be constructed for the y and z axes simply by replacing occurrences of x with y and z, respectively.

Let $\hat{\mathbf{q}}_{xk}$ represent the x-axis coordinate predictions for the data points in the *k*th segment, $\boldsymbol{\varphi}_k$ is the regressor matrix composed of B-spline basis function evaluations, and $\mathbf{p}_{xk}$ is a vector of the unknown x-axis coordinates of the control points in the *k*th segment. Note that the regressor matrix is the same for the y and z axes as well. Aggregating all of the x-axis coordinate predictions in the first (k = 1) NURBS segment results in the following system of equations:

$$\underbrace{\begin{bmatrix} \hat{q}_{x1,0} \\ \hat{q}_{x1,1} \\ \vdots \\ \hat{q}_{x1,n_1} \end{bmatrix}}_{\hat{\mathbf{q}}_{x1}} = \underbrace{\begin{bmatrix} N_{0,p}(\bar{u}_{1,0}) & N_{1,p}(\bar{u}_{1,0}) & \cdots & N_{n_1,p}(\bar{u}_{1,0}) \\ N_{0,p}(\bar{u}_{1,1}) & N_{1,p}(\bar{u}_{1,1}) & \cdots & N_{n_1,p}(\bar{u}_{1,1}) \\ \vdots & \vdots & & \vdots \\ N_{0,p}(\bar{u}_{1,n_1}) & N_{1,p}(\bar{u}_{1,n_1}) & \cdots & N_{n_1,p}(\bar{u}_{1,n_1}) \end{bmatrix}}_{\boldsymbol{\varphi}_1} \underbrace{\begin{bmatrix} P_{x1,0} \\ P_{x1,1} \\ \vdots \\ P_{x1,n_1} \end{bmatrix}}_{\mathbf{p}_{x1}} \tag{3.7}$$

For all other segments (k > 1), the first data point coincides with the last data point of the previous segment. Equivalently, the first control point coincides with the last control point of the previous segment, which naturally enforces G^0 continuity between the segments.

Therefore, all subsequent segments can drop the first data point from the formulation as follows:

$$\underbrace{\begin{bmatrix} \hat{q}_{xk,1} \\ \vdots \\ \hat{q}_{xk,n_k} \end{bmatrix}}_{\hat{\mathbf{q}}_{xk}} = \underbrace{\begin{bmatrix} N_{0,p}(\bar{u}_{k,1}) & N_{1,p}(\bar{u}_{k,1}) & \cdots & N_{n_k,p}(\bar{u}_{k,1}) \\ \vdots & \vdots & & \vdots \\ N_{0,p}(\bar{u}_{k,n_k}) & N_{1,p}(\bar{u}_{k,n_k}) & \cdots & N_{n_k,p}(\bar{u}_{k,n_k}) \end{bmatrix}}_{\boldsymbol{\varphi}_k} \underbrace{\begin{bmatrix} P_{xk,1} \\ \vdots \\ P_{xk,n_k} \end{bmatrix}}_{\mathbf{p}_{xk}} \tag{3.8}$$

Aggregating the equations from all segments results in the following system of equations:

$$\underbrace{\begin{bmatrix} \hat{\mathbf{q}}_{x1} \\ \hat{\mathbf{q}}_{x2} \\ \vdots \\ \hat{\mathbf{q}}_{xN} \end{bmatrix}}_{\hat{\mathbf{Q}}_x} = \underbrace{\begin{bmatrix} \boldsymbol{\varphi}_1 & 0 & \cdots & 0 \\ 0 & \boldsymbol{\varphi}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\varphi}_N \end{bmatrix}}_{\mathbf{\Phi}} \underbrace{\begin{bmatrix} \mathbf{p}_{x1} \\ \mathbf{p}_{x2} \\ \vdots \\ \mathbf{p}_{xN} \end{bmatrix}}_{\mathbf{P}_x} \tag{3.9}$$

Since the total number of unique control points equals the number of data points, which is $M+1$, the regressor matrix is a square block diagonal matrix with dimensions $M+1$ x $M+1$.

The error between the actual x-axis coordinates $\mathbf{Q}_x$ and the predicted x-coordinates $\hat{\mathbf{Q}}_x$ is:

$$\mathbf{e}_x = \mathbf{Q}_x - \hat{\mathbf{Q}}_x = \mathbf{Q}_x - \mathbf{\Phi}\mathbf{P}_x \tag{3.10}$$

The objective function to be minimized is:

$$J_x = \frac{1}{2}\mathbf{e}_x{}^T\mathbf{e}_x = (\mathbf{Q}_x - \mathbf{\Phi}\mathbf{P}_x)^T(\mathbf{Q}_x - \mathbf{\Phi}\mathbf{P}_x) \tag{3.11}$$

Next, the optimization problem is constrained by position, tangent and curvature continuity constraints at the toolpath's start and end points as well as the segment junctions. To achieve position G^0 continuity at NURBS segment boundaries, the first control point must equal the last control point of the previous segment. Since this constraint has already been incorporated into the formulation of the regressor matrix, it is not required to include these equations in the constraints. However position constraints at the start and end of the toolpath must be imposed such that the first and last control points of the toolpath are equal to the first and last data points, respectively.

$$\underbrace{\begin{bmatrix} q_{x1,0} \\ q_{xN,n_N} \end{bmatrix}}_{\xi_x^{(0)}} = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}}_{\mathbf{L}^{(0)}} \underbrace{\begin{bmatrix} \mathbf{p}_{x1} \\ \vdots \\ \mathbf{p}_{xN} \end{bmatrix}}_{\mathbf{P}_x} \tag{3.12}$$

Tangent and curvature continuity require that the derivatives with respect to arc-length, *s*, evaluated at the segment boundaries must be equal.

$$\begin{aligned} \left.\frac{d\mathbf{C}_{k-1}}{ds}\right|_{u=1} &= \left.\frac{d\mathbf{C}_k}{ds}\right|_{u=0} \\ \left.\frac{d^2\mathbf{C}_{k-1}}{ds^2}\right|_{u=1} &= \left.\frac{d^2\mathbf{C}_k}{ds^2}\right|_{u=0} \end{aligned} \tag{3.13}$$

Since chord length parameterization was employed to obtain the $\bar{u}$ values and knots in the knot vectors, the spline parameter, *u*, is generally not equal to the arc-length, *s*. Hence, $ds \neq du$. Moreover, each segment has a different parameterization since the chord lengths are not uniform throughout the entire toolpath. A general approach to satisfy Equation (3.13) is to use *beta-constraints*, which were derived in Section 2.3. Equation (2.4) serves as the mathematical basis for formulating the tangent and curvature continuity constraints at the junctions of segments with different parameterizations. First and second order beta-constraints at the *k-1*th and *k*th segment boundary are expressed in Equation (3.14).

$$\begin{aligned} \left.\frac{d\mathbf{C}_k}{du}\right|_{u=0} &= \beta 1 \cdot \left.\frac{d\mathbf{C}_{k-1}}{du}\right|_{u=1} \\ \left.\frac{d^2\mathbf{C}_k}{du^2}\right|_{u=0} &= \beta 1^2 \cdot \left.\frac{d^2\mathbf{C}_{k-1}}{du^2}\right|_{u=1} + \beta 2 \cdot \left.\frac{d\mathbf{C}_{k-1}}{du}\right|_{u=1} \end{aligned} \tag{3.14}$$

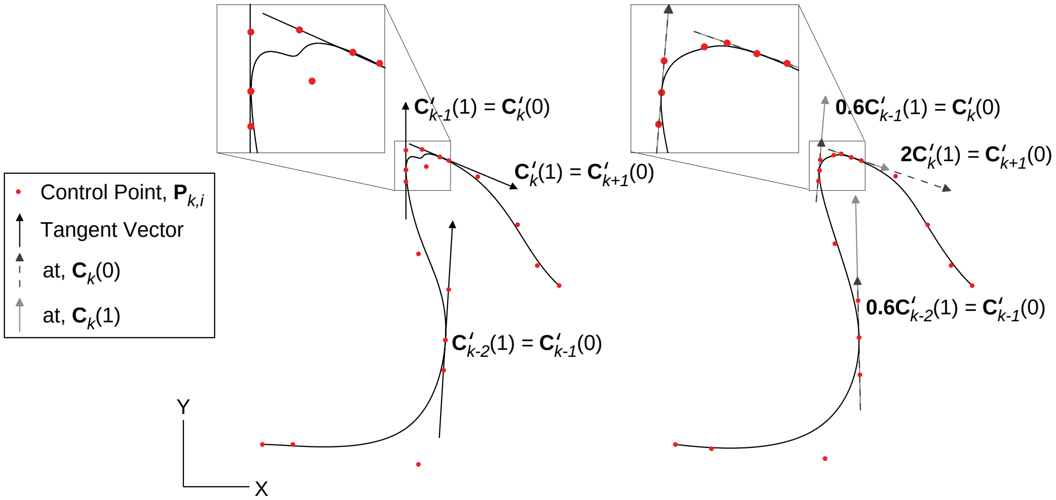

Figure 3-5. Effect of tangent beta-constraint value, β1, on curve fitting.

Beta values, $\beta1$ and $\beta2$, are scalar shape parameters that influence how adjacent segments join smoothly. Note that by setting $\beta1 = 1$ and $\beta2 = 0$, Equation (3.14) reduces to first and second order parametric continuity constraints, which are normally used in arc-length parameterization methods. However, since the parameterization is not with respect to the arc-length, imposing pure parametric continuity constraints can result in unwanted oscillations in the fitted curve, as illustrated in Figure 3-5. In comparison, imposing geometric continuity constraints with varying beta values result in a smooth curve without oscillations. Beta values are chosen such that segments join smoothly without oscillations. Each segment is assigned its own shape parameters. For G^1 continuity, the first derivative vector evaluated at the start point of the *k*th NURBS segment must be a positive multiple, $\beta1$, of the first derivative vector evaluated at the end point of the previous segment. Let $\mathbf{C}_k^{(1)}(u)$ represent the first parametric derivative of the *k*th segment. Then the first derivative beta constraint can be written as:

$$0 = \beta1_{k-1}\mathbf{C}_{k-1}^{(1)}(1) - \mathbf{C}_k^{(1)}(0), \qquad \beta1_{k-1} > 0 \tag{3.15}$$

Similarly, the beta constraint for G^2 continuity states that a linear combination of the second derivative vectors evaluated at the boundary point of the *k-1*th and *k*th segments is a multiple, $\beta2$, of the first derivative vector of the *k-1*th segment. Let $\mathbf{C}_k^{(2)}(u)$ represent the

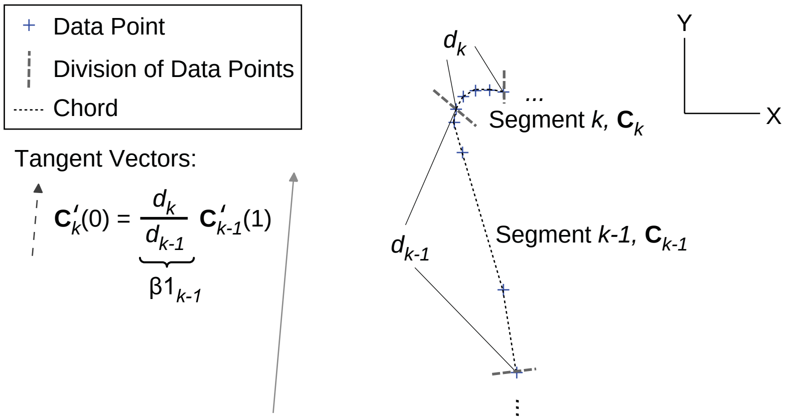
+ Data Point
Division of Data Points
Chord
Tangent Vectors:
$\mathbf{C}'_k(0) = \underbrace{\frac{d_k}{d_{k-1}}}_{\beta 1_{k-1}} \mathbf{C}'_{k-1}(1)$
d_k
...
Segment k, $\mathbf{C}_k$
d_{k-1}
Segment k-1, $\mathbf{C}_{k-1}$
Y
X

Figure 3-6. Rule of thumb to calculate tangent beta value, β1.

second parametric derivative of the *k*th segment. The second derivative beta constraint can then be expressed with the following equation:

$$0 = \beta 2_{k-1}\mathbf{C}_{k-1}^{(1)}(1) + \beta 1_{k-1}^2 \mathbf{C}_{k-1}^{(2)}(1) - \mathbf{C}_k^{(2)}(0) \tag{3.16}$$

Satisfying these constraints ensures that the segments are geometrically continuous at segment boundaries, despite differences in the parameterizations between adjacent segments. A rule of thumb that works well in selecting beta parameters is to use the ratios of the summed chord lengths, which was denoted as d_k in Equation (3.4), for β1 and then setting $\beta 2$ to zero. That is,

$$\beta 1_{k-1} = \frac{d_k}{d_{k-1}}, \qquad \beta 2_{k-1} = 0 \tag{3.17}$$

As illustrated in Figure 3-6, if data points are clustered close together, then the tangent vector should be correspondingly shorter, and if data points are spread out, then the tangent vector should be correspondingly longer. However, some trial and error may be necessary to reduce oscillatory behaviour in the fitted spline. Using a ratio of knot spans is another option, but the results are not always predictable because the parameterizations are normalized to be between zero and one. In general, a ratio based on the chord lengths is used in this thesis, and manually adjusted as necessary.

To construct the linear equations for the tangent and curvature boundary conditions, first and second order beta-constraints are generated for each segment boundary. First, the *l*th

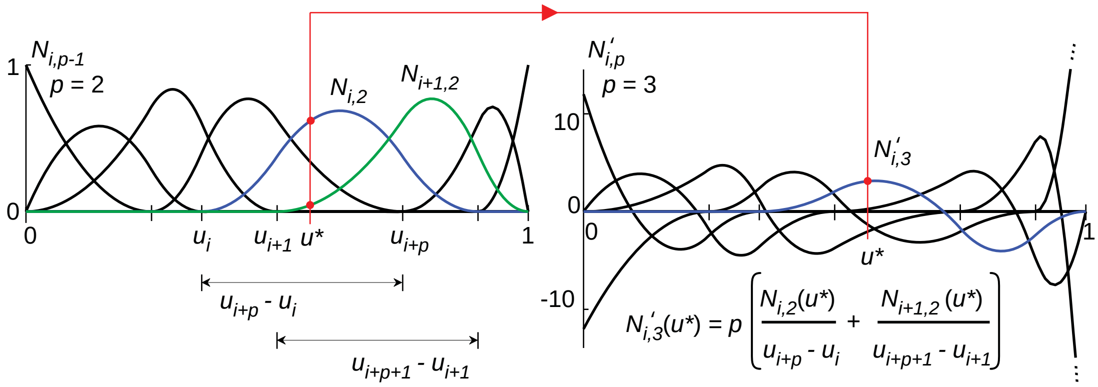

Figure 3-7. Evaluation of the first derivative B-spline basis function (*p* = 3).

derivative of the *k*th segment can be obtained by computing the derivatives of the B-spline basis functions recursively with Equation (3.18). The first derivative B-spline basis functions, $N'_{i.3}$, are illustrated in Figure 3-7.

$$N_{i,p}^{(l)}(u) = p\left(\frac{N_{i,p-1}^{(l-1)}(u)}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(l-1)}(u)}{u_{i+p+1} - u_{i+1}}\right) \tag{3.18}$$

$$\mathbf{C}_k^{(l)}(u) = \sum_{i=0}^{n_k} N_{i,p}^{(l)}(u)\mathbf{P}_{k,i} = \left[N_{0,p}^{(l)}(u) \quad \cdots \quad N_{n_k,p}^{(l)}(u)\right]\begin{bmatrix}\mathbf{P}_{k,0}\\ \vdots \\ \mathbf{P}_{k,n_k}\end{bmatrix} \tag{3.19}$$

The parametric derivative vectors are thus calculated as a linear combination of the control points, as show in Equation (3.19). Using Equations (3.15) and (3.19), the tangent continuity constraint between the *k-1*th and *k*th segment can be expressed with Equation (3.20). Note that for all subsequent constraint equations that involve the *k-1*th and *k*th segments, if $N_{i,p}^{(1)}$ is evaluated at one, then it is computed with the *k-1*th knot vector, $\mathbf{U}_{k-1}$. If $N_{i,p}^{(1)}$ is evaluated at zero, then it is computed with the *k*th knot vector $\mathbf{U}_k$.

$$0 = \underbrace{\begin{bmatrix} \beta 1_{k-1} N_{0,p}^{(1)}(1) & \beta 1_{k-1} N_{1,p}^{(1)}(1) & \cdots & \beta 1_{k-1} N_{n_{k-1}-1,p}^{(1)}(1) \\ \beta 1_{k-1} N_{n_{k-1},p}^{(1)}(1) - N_{0,p}^{(1)}(0) & -N_{i,p}^{(1)}(0) & \cdots & -N_{n_k-1,p}^{(1)}(0) \quad -N_{n_k,p}^{(1)}(0) \end{bmatrix}}_{\mathbf{L}_k^{(1)}} \begin{bmatrix}\mathbf{p}_{xk-1}\\ \mathbf{p}_{xk}\end{bmatrix} \tag{3.20}$$

If the toolpath is closed, then it may be desirable to also include a tangent continuity constraint at the start and end of the toolpath. In this case, the B-spline basis functions, $N_{i,p}^{(1)}(0)$, are evaluated with the first segment's knot vector, while $N_{i,p}^{(1)}(1)$ are evaluated with that of the last segment. The following equation can be inserted into the constraints:

$$0 = \underbrace{\begin{bmatrix} -N_{0,p}^{(1)}(0) & \cdots & -N_{n_1,p}^{(1)}(0) & 0 & \cdots & 0 & \beta1_N N_{0,p}^{(1)}(1) & \cdots & \beta1_N N_{n_N,p}^{(1)}(1) \end{bmatrix}}_{\mathbf{L}_{closed}^{(1)}} \begin{bmatrix} \mathbf{p}_{x1} \\ \vdots \\ \mathbf{p}_{xN} \end{bmatrix} \qquad (3.21)$$

Aggregating all the tangent G^1 continuity constraints into one matrix gives:

$$\underbrace{\begin{bmatrix} 0_{N-1(+1)\times 1} \end{bmatrix}}_{\xi_x^{(1)}} = \underbrace{\begin{bmatrix} [\mathbf{L}_2^{(1)}]_{1\times 2n+1} & 0_{1\times M-2n} & & \\ 0_{1\times n} & [\mathbf{L}_3^{(1)}]_{1\times 2n+1} & 0_{1\times M-3n} & \\ \ddots & \ddots & \ddots & \\ 0_{1\times (k-2)n} & [\mathbf{L}_k^{(1)}]_{1\times 2n+1} & 0_{1\times M-kn} & \\ \ddots & \ddots & \ddots & \\ 0_{1\times n(N-3)} & [\mathbf{L}_{N-1}^{(1)}]_{1\times 2n+1} & 0_{1\times M-n(N-1)} & \\ 0_{1\times n(N-4)} & [\mathbf{L}_N^{(1)}]_{1\times 2n+1} & & \\ (& [\mathbf{L}_{closed}^{(1)}]_{1\times M+1} &) & \end{bmatrix}}_{\mathbf{L}^{(1)}} \underbrace{\begin{bmatrix} \mathbf{p}_{x1} \\ \mathbf{p}_{x2} \\ \vdots \\ \mathbf{p}_{xk-1} \\ \mathbf{p}_{xk} \\ \vdots \\ \mathbf{p}_{xN-1} \\ \mathbf{p}_{xN} \end{bmatrix}}_{\mathbf{P}_x} \qquad (3.22)$$

There are $N-1$ segment boundaries. Hence there are $N-1$ tangent constraints and one optional constraint for closed toolpaths, which is marked with the curved brackets.

Similarly, the curvature continuity constraint matrix is formulated with Equations (3.16) and (3.19). The curvature continuity constraint at the *k-1*th and *k*th segment boundary can be expressed with the following equation:

$$0 = \underbrace{\begin{matrix} [\ \beta1_{k-1}^2 N_{0,p}^{(2)}(1) + \beta2_{k-1} N_{0,p}^{(1)}(1) \quad \beta1_{k-1}^2 N_{1,p}^{(2)}(1) + \beta2_{k-1} N_{1,p}^{(1)}(1) \quad \cdots \\ \beta1_{k-1}^2 N_{n_{k-1},p}^{(2)}(1) + \beta2_{k-1} N_{n_{k-1},p}^{(1)}(1) - N_{0,p}^{(2)}(0) \quad -N_{1,p}^{(2)}(0) \quad \cdots \quad -N_{n_k,p}^{(2)}(0)\ ] \end{matrix}}_{\mathbf{L}_k^{(2)}} \begin{bmatrix} \mathbf{p}_{xk-1} \\ \mathbf{p}_{xk} \end{bmatrix} \qquad (3.23)$$

For a closed curve, the curvature continuity constraint at the start and end points yields the equation:

$$0 = \underbrace{\begin{bmatrix} N^{(2)}_{0,p}(0) \quad \cdots \quad N^{(2)}_{n_1,p}(0) \quad 0 \quad \cdots \quad 0 \\ -\beta 1^2_N N^{(2)}_{0,p}(1) - \beta 2_N N^{(1)}_{0,p}(1) \quad \cdots \quad -\beta 1^2_N N^{(2)}_{n_N,p}(1) - \beta 2_N N^{(1)}_{n_N,p}(1) \end{bmatrix}}_{\mathbf{L}^{(2)}_{closed}} \begin{bmatrix} \mathbf{p}_{x1} \\ \vdots \\ \mathbf{p}_{xN} \end{bmatrix} \tag{3.24}$$

Aggregating all the curvature continuity constraints and the optional closed loop constraint into one matrix gives:

$$\underbrace{\begin{bmatrix} 0_{N-1(+1)\times 1} \end{bmatrix}}_{\xi^{(1)}_x} = \underbrace{\begin{bmatrix} [\mathbf{L}^{(2)}_2]_{1\times 2n+1} \quad 0_{1\times M-2n} \\ 0_{1\times n} \quad [\mathbf{L}^{(2)}_3]_{1\times 2n+1} \quad 0_{1\times M-3n} \\ \ddots \qquad \ddots \qquad \ddots \\ 0_{1\times (k-2)n} \quad [\mathbf{L}^{(2)}_k]_{1\times 2n+1} \quad 0_{1\times M-kn} \\ \ddots \qquad \ddots \qquad \ddots \\ 0_{1\times (N-3)n} \quad [\mathbf{L}^{(2)}_{N-1}]_{1\times 2n+1} \quad 0_{1\times M-(N-1)n} \\ 0_{1\times n(N-4)} \quad [\mathbf{L}^{(2)}_N]_{1\times 2n+1} \\ (\qquad [\mathbf{L}^{(2)}_{closed}]_{1\times M+1} \qquad) \end{bmatrix}}_{\mathbf{L}^{(1)}} \underbrace{\begin{bmatrix} \mathbf{p}_{x1} \\ \mathbf{p}_{x2} \\ \vdots \\ \mathbf{p}_{xk-1} \\ \mathbf{p}_{xk} \\ \vdots \\ \mathbf{p}_{xN-1} \\ \mathbf{p}_{xN} \end{bmatrix}}_{\mathbf{P}_x} \tag{3.25}$$

The resultant constraint equations are:

$$\underbrace{\begin{bmatrix} \xi^{(0)}_x \\ \xi^{(1)}_x \\ \xi^{(2)}_x \end{bmatrix}}_{\xi_x} = \underbrace{\begin{bmatrix} [\mathbf{L}^{(0)}]_{2\times M+1} \\ [\mathbf{L}^{(1)}]_{N-1(+1)\times M+1} \\ [\mathbf{L}^{(2)}]_{N-1(+1)\times M+1} \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} \mathbf{p}_{x1} \\ \vdots \\ \mathbf{p}_{xN} \end{bmatrix}}_{\mathbf{P}_x} \tag{3.26}$$

Hence, the x-axis coordinates of the unknown control points, $\mathbf{P}_x$, are obtained by solving the following constrained linear quadratic optimization problem:

$$\min_{\mathbf{P}_x} J_x = \min_{\mathbf{P}_x} \frac{1}{2}(\mathbf{Q}_x - \mathbf{\Phi}\mathbf{P}_x)^T(\mathbf{Q}_x - \mathbf{\Phi}\mathbf{P}_x) \qquad \text{Subject to: } \; \mathbf{L}\cdot\mathbf{P}_x = \xi_x \tag{3.27}$$

For an *N*-segment toolpath, there are 2 position constraints, $N-1$ tangent constraints and $N-1$ curvature constraints, and an optional 2 more constraints if the toolpath is closed.

The total number of constraints is thus 2*N*(+2), where the number in parentheses represents the additional closed toolpath constraints. Using Lagrange multipliers, $\mathbf{\Lambda} = [\lambda_0 \quad \lambda_1 \quad \cdots \quad \lambda_{2N(+2)}]^T$, the augmented objective function is constructed as:

$$\min_{\mathbf{P}_x,\mathbf{\Lambda}} J'_x = \min_{\mathbf{P}_x,\mathbf{\Lambda}} \frac{1}{2}(\mathbf{Q}_x - \mathbf{\Phi}\mathbf{P}_x)^T(\mathbf{Q}_x - \mathbf{\Phi}\mathbf{P}_x) + \mathbf{\Lambda}^T(\mathbf{L}\cdot\mathbf{P}_x - \xi_x) \tag{3.28}$$

Differentiating Equation (3.28) with respect to $\mathbf{P}_x$ and $\mathbf{\Lambda}$, then setting the partial derivatives to zero yields the linear system of equations:

$$\begin{bmatrix} \mathbf{\Phi}^T\mathbf{\Phi} & \mathbf{L}^T \\ \mathbf{L} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}_x \\ \mathbf{\Lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}^T\mathbf{Q}_x \\ \xi_x \end{bmatrix} \tag{3.29}$$

Solving the linear system in Equation (3.29) yields the x-axis coordinates of the unknown control points. Y and z coordinates are obtained by replacing the values of $\mathbf{P}_x$, $\mathbf{Q}_x$, and ξ_x, accordingly. The optimally placed control points, along with the computed knot vectors and weights, define an *N*-segment NURBS toolpath which is subsequently interpolated at the control loop sampling frequency to generate the toolpath reference trajectory.

3.3 NURBS Toolpath Interpolation

3.3.1 Segment Arc-length Calculation

For the interpolator to realize the motion smoothly, it is important to calculate the arc-length of each segment accurately. An inaccurate estimate for the arc-length will result in either under- or over-shooting the desired end point, thus causing trajectory discontinuities between segments. The total segment arc-length is an integration of infinitesimally small arc-lengths, $L = \int ds$. Using Pythagoras' theorem, ds is the hypotenuse of infinitesimally small displacements in the x, y and z directions. Differentiating with respect to the spline parameter, *u*, the arc-length differential can be expressed as follows:

$$\frac{ds}{du} = \sqrt{\frac{dx}{du}^2 + \frac{dy}{du}^2 + \frac{dz}{du}^2} = \sqrt{x'(u)^2 + y'(u)^2 + z'(u)^2} \tag{3.30}$$

Thus, the total arc-length is formulated as follows:

$$L = \int_a^b \sqrt{x'(u)^2 + y'(u)^2 + z'(u)^2}\,du = \int_a^b f(u)du, \qquad a \le u \le b \tag{3.31}$$

There currently exists no analytical solution for Equation (3.31). However, using Simpson's rule with an adaptive bisection technique, the arc-length can be calculated numerically within a specific tolerance of its true value. This procedure was also employed by Lei et al. [18]. First, an approximation of the arc-length is performed by evaluating the integrand, $f(u)$, at both end points of the spline parameter interval, which is denoted as $[a,b]$, and its midpoint (c), and applying Simpson's rule to obtain the arc-length estimate $l(a,b)$. The midpoint is calculated as $c = (a+b)/2$. The step size is $h = (b-a)/2$.

$$Simpson's \quad Rule: \qquad l(a,b) = \frac{h}{3}\big(f(a) + 4f(c) + f(b)\big) \tag{3.32}$$

Next, the interval $[a,b]$ is split into two equal sized intervals, denoted as $[a_1,b_1]$ and $[a_2,b_2]$, and Simpson's rule is applied on both subintervals to obtain the lengths $l(a_1,b_1)$ and $l(a_2,b_2)$. Given a specified tolerance, ε, if the condition in Equation (3.33) is satisfied, then the approximation is within the given tolerance of the true arc-length.

$$|\, l(a_1,b_1) + l(a_2,b_2) - l(a,b)\,|/10 < \varepsilon \tag{3.33}$$

A proof of this statement is provided by Mathews and Fink in [38]. If the condition is not satisfied, then the subintervals are further refined by dividing them into two, halving the tolerance value, and reapplying Simpson's rule. This procedure iterates until all subintervals satisfy the tolerance, which is guaranteed to occur in a finite number of subdivisions, assuming that the fourth derivative of the integrand, $f^{(4)}(u)$, is continuous over the interval $[a,b]$ [38]. Each refinement reduces the error by approximately a factor of 1/16, as shown in [38]. The total arc-length is calculated by summing up the subinterval lengths. Moreover, a cumulative summation of the subinterval lengths is also performed, in order to produce the spline parameter and arc-length pairs (u_i, s_i). In this notation, i is an integer value between zero and the total number of subintervals, s_i is the sum of the arc-lengths up to and including

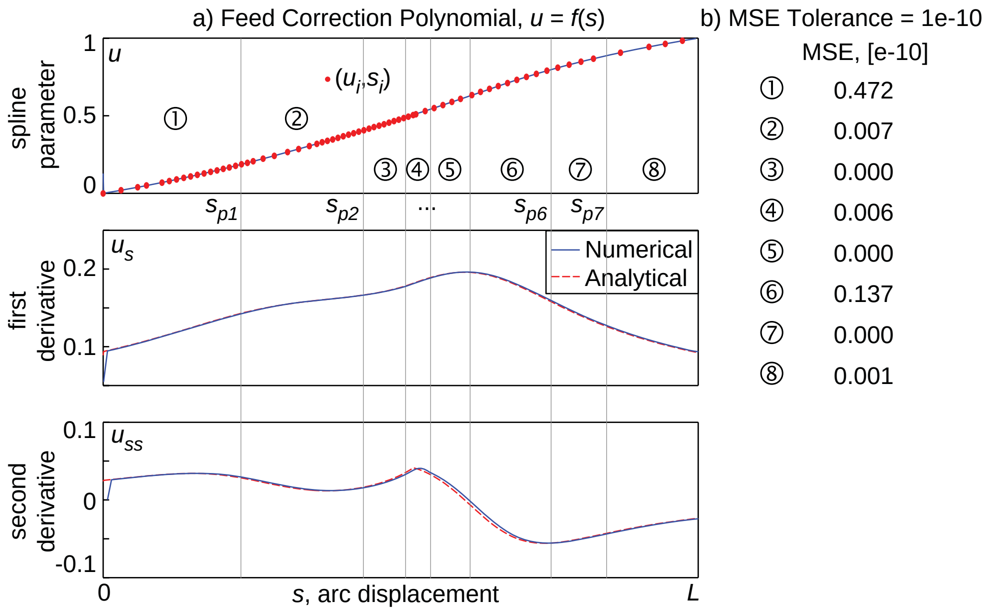

Figure 3-8. a) Feed correction polynomial fitting and comparison of its analytical and numerical derivatives. There are 8 polynomials. b) Mean squared error (MSE) of curve fit for each polynomial.

subinterval i, and u_i is the corresponding spline parameter value. N_Σ is the number of points used to numerically integrate the segment's arc-length. These points are used in fitting the feed correction polynomials.

3.3.2 Feed Correction Polynomial

In general, NURBS toolpath parameterization does not yield a perfectly arc-length parameterized curve, which results in unwanted fluctuations in the feedrate, (i.e. tangential velocity), when the spline parameter is interpolated at constant increments. Feedrate fluctuations cause unsmooth tool motion, which causes visible feed marks on the machined part. Moreover, small discontinuities in the tangential velocity are magnified in the acceleration and jerk profiles. High acceleration and jerk may result in saturation of the motor actuators and excitation of the machine tool's structural modes, which have the effect of degrading the tracking performance. Therefore, feedrate fluctuations should be avoided. In order to correct this problem, a scalar valued function, $u = f(s)$, is employed to map the desired arc displacement s to the correct spline parameter u. In this work, this function is referred to as the *feed correction polynomial*. Lei et al. [18] employed a similar reparameterization scheme using cubic Hermite splines. Here, a 7th order polynomial is used to approximate the (u_i, s_i) data, as proposed in [10]. A *7*th order polynomial is chosen such that boundary conditions on the position, and first and second derivatives at the start and end points of the function can be imposed, which requires at the least a *5*th order polynomial. The extra two degrees of freedom are to better approximate the data without introducing polynomial "wiggle". However, in certain cases it was found that a single *7*th order polynomial was insufficient to capture the relationship between the spline parameter and the arc displacement, and results were sometimes completely erroneous. It was noticed that, in general, when large changes in the spline parameter only result in small changes in the arc displacement, the curve fitting would run into numerical instability issues. Rather than increasing the order of the polynomial, it was found that multiple feed correction polynomials could approximate the relationship better. As an extension to the earlier work in [10], this chapter presents a procedure to connect multiple *7*th order polynomials while maintaining first and second derivative continuity throughout the curve fitting, as illustrated in Figure 3-8. Segment ①'s polynomial is used to calculate the spline parameter for arc displacements in the interval $[0, s_{p1}]$; segment ②'s polynomial is used to obtain the spline parameter for arc displacements in the interval $[s_{p1}, s_{p2}]$, and so on. When a single curve fails to achieve an acceptable value for the mean squared error (MSE) of approximation, then the data points are split in half and two curves are fitted. It was found that splitting the point data and fitting multiple curves to them reduced this error.

To start, a single curve is approximated to the (u_i, s_i) data in a least squares sense. The feed correction polynomial has the form:

$$\hat{u} = f(s) = A_0 s^7 + A_1 s^6 + A_2 s^5 + A_3 s^4 + A_4 s^3 + A_5 s^2 + A_6 s + A_7 \qquad (3.34)$$

In order to avoid ill-conditioning, the arc-length data $\mathbf{s} = [s_0, \ldots, s_{N_\Sigma}]$ is normalized to be between 0 and 1, by defining $\sigma_i = (s_i - s_0)/(s_{N_\Sigma} - s_0)$ for $i = 0,1,\ldots,N_\Sigma$, thus $\boldsymbol{\sigma} = [0, \sigma_1, \ldots, \sigma_{N_\Sigma - 1}, 1]$. Hence, the feed correction polynomial and its derivatives, in terms of normalized variables and coefficients, are obtained as follows:

$$\left.\begin{aligned}\hat{u} &= f(s) = \alpha_0\sigma^7 + \alpha_1\sigma^6 + \alpha_2\sigma^5 + \alpha_3\sigma^4 + \alpha_4\sigma^3 + \alpha_5\sigma^2 + \alpha_6\sigma + \alpha_7 \\ \hat{u}_s &= f'(s) = 7\alpha_0\sigma^6 + 6\alpha_1\sigma^5 + 5\alpha_2\sigma^4 + 4\alpha_3\sigma^3 + 3\alpha_4\sigma^2 + 2\alpha_5\sigma + \alpha_6 \\ \hat{u}_{ss} &= f''(s) = 42\alpha_0\sigma^5 + 30\alpha_1\sigma^4 + 20\alpha_2\sigma^3 + 12\alpha_3\sigma^2 + 6\alpha_4\sigma + 2\alpha_5\end{aligned}\right\} \tag{3.35}$$

The spline parameter predictions of $\hat{\mathbf{u}} = [\hat{u}_0,\ldots,\hat{u}_{N_\Sigma}]$, calculated with the feed correction polynomial, are stacked together in matrix form, as shown in Equation (3.36). Here, $\boldsymbol{\Phi}$ denotes the regressor matrix, and $\boldsymbol{\theta}$ is the vector of coefficients.

$$\underbrace{\begin{bmatrix}\hat{u}_0 \\ \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_N\end{bmatrix}}_{\hat{\mathbf{u}}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \sigma_1^7 & \sigma_1^6 & \sigma_1^5 & \sigma_1^4 & \sigma_1^3 & \sigma_1^2 & \sigma_1 & 1 \\ \sigma_2^7 & \sigma_2^6 & \sigma_2^5 & \sigma_2^4 & \sigma_2^3 & \sigma_2^2 & \sigma_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}}_{\boldsymbol{\Phi}} \underbrace{\begin{bmatrix}\alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_6 \\ \alpha_7\end{bmatrix}}_{\boldsymbol{\theta}} \tag{3.36}$$

Zero, first, and second order boundary conditions are imposed on the feed correction polynomial, in order to preserve continuity between the connecting segments. The first and second order derivatives are evaluated at the boundary points with the expressions of u_s and u_{ss}, obtained by applying the chain rule:

$$\begin{aligned}&u_s = \frac{du}{ds} = [g(u)]^{-\frac{1}{2}} \quad , \quad u_{ss} = \frac{d^2u}{ds^2} = -\frac{1}{2}[g(u)]^{-2} g'(u) \\ &g(u) = x'(u)^2 + y'(u)^2 + z'(u)^2 \\ &g'(u) = 2[x'(u)x''(u) + y'(u)y''(u) + z'(u)z''(u)]\end{aligned} \tag{3.37}$$

Denoting the *1*st and *2*nd derivatives evaluated at the start and end points (u^{init} and u^{final}) using Equation (3.37) as u_s^{init}, u_{ss}^{init}, u_s^{final}, and u_{ss}^{final}, and $\Delta s = s_{N_\Sigma} - s_0$, the resulting zero, first, and second order derivative constraints can be written in matrix form as:

$$\underbrace{\begin{bmatrix} u^{init} \\ \Delta s \cdot u_s^{init} \\ \Delta s^2 \cdot u_{ss}^{init} \\ u^{final} \\ \Delta s \cdot u_s^{final} \\ \Delta s^2 \cdot u_{ss}^{final} \end{bmatrix}}_{\xi} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 42 & 30 & 20 & 12 & 6 & 2 & 0 & 0 \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_6 \\ \alpha_7 \end{bmatrix}}_{\boldsymbol{\theta}} \tag{3.38}$$

Using the method of Lagrange multipliers, it can be shown that the constrained optimization problem, which minimizes the squared error $\mathbf{e}^T\mathbf{e} = (\mathbf{u} - \boldsymbol{\Phi\theta})^T(\mathbf{u} - \boldsymbol{\Phi\theta})$, such that $\xi = \mathbf{L}\boldsymbol{\theta}$, results in the system of linear equations in Equation (3.39). This result was previously derived in Equations (3.27)-(3.29):

$$\begin{bmatrix} \boldsymbol{\Phi}^T\boldsymbol{\Phi} & \mathbf{L}^T \\ \mathbf{L} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\Lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^T\mathbf{u} \\ \xi \end{bmatrix} \tag{3.39}$$

Above, $\boldsymbol{\Lambda} = [\lambda_0, \ldots, \lambda_5]^T$ is the vector of Lagrange multipliers. Solving Equation (3.39) for $\boldsymbol{\theta}$ yields the normalized coefficients α_i, which minimize the error between the predicted and true spline parameter values. If $s_0 = 0$, then the original coefficients A_i in Equation (3.34) are solved by de-normalizing the α_i coefficients, resulting in $A_0 = \alpha_0 / \Delta s^7$, $A_1 = \alpha_1 / \Delta s^6$, …, $A_7 = \alpha_7$. Otherwise the original coefficients are obtained by substituting $\sigma = \left(\dfrac{s - s_0}{\Delta s}\right)$ into the normalized feed correction polynomial in Equation (3.35), as follows:

$$f(s) = \alpha_0\left(\frac{s - s_0}{\Delta s}\right)^7 + \alpha_1\left(\frac{s - s_0}{\Delta s}\right)^6 + \cdots + \alpha_5\left(\frac{s - s_0}{\Delta s}\right)^2 + \alpha_6\left(\frac{s - s_0}{\Delta s}\right) + \alpha_7 \tag{3.40}$$

Expanding Equation (3.40) and grouping like terms yields the following expressions for the coefficient of each power term:

$$
\begin{aligned}
s^7 : A_0 &= \frac{\alpha_0}{\Delta s^7} \\
s^6 : A_1 &= \frac{\alpha_1}{\Delta s^6} - 7s_0 \frac{\alpha_0}{\Delta s^7} \\
s^5 : A_2 &= \frac{\alpha_2}{\Delta s^5} - 6s_0 \frac{\alpha_1}{\Delta s^6} + 21s_0^2 \frac{\alpha_0}{\Delta s^7} \\
s^4 : A_3 &= \frac{\alpha_3}{\Delta s^4} - 5s_0 \frac{\alpha_2}{\Delta s^5} + 15s_0^2 \frac{\alpha_1}{\Delta s^6} - 35s_0^3 \frac{\alpha_0}{\Delta s^7} \\
s^3 : A_4 &= \frac{\alpha_4}{\Delta s^3} - 4s_0 \frac{\alpha_3}{\Delta s^4} + 10s_0^2 \frac{\alpha_2}{\Delta s^5} - 20s_0^3 \frac{\alpha_1}{\Delta s^6} + 35s_0^4 \frac{\alpha_0}{\Delta s^7} \\
s^2 : A_5 &= \frac{\alpha_5}{\Delta s^2} - 3s_0 \frac{\alpha_4}{\Delta s^3} + 6s_0^2 \frac{\alpha_3}{\Delta s^4} - 10s_0^3 \frac{\alpha_2}{\Delta s^5} + 15s_0^4 \frac{\alpha_1}{\Delta s^6} - 21s_0^5 \frac{\alpha_0}{\Delta s^7} \\
s : A_6 &= \frac{\alpha_6}{\Delta s} - 2s_0 \frac{\alpha_5}{\Delta s^2} + 3s_0^2 \frac{\alpha_4}{\Delta s^3} - 4s_0^3 \frac{\alpha_3}{\Delta s^4} + 5s_0^4 \frac{\alpha_2}{\Delta s^5} - 6s_0^5 \frac{\alpha_1}{\Delta s^6} + 7s_0^6 \frac{\alpha_0}{\Delta s^7} \\
1 : A_7 &= \alpha_7 - s_0 \frac{\alpha_6}{\Delta s} + s_0^2 \frac{\alpha_5}{\Delta s^2} - s_0^3 \frac{\alpha_4}{\Delta s^3} + s_0^4 \frac{\alpha_3}{\Delta s^4} - s_0^5 \frac{\alpha_2}{\Delta s^5} + s_0^6 \frac{\alpha_1}{\Delta s^6} - s_0^7 \frac{\alpha_0}{\Delta s^7}
\end{aligned}
\tag{3.41}
$$

It can be seen that the numerical values in front of the normalized coefficients are in fact entries of Pascal's triangle with alternating signs, which is a result of the binomial expansions of $(s - s_0)^n$ for $n = 1,2,\ldots,7$. A generalized formula for the de-normalized coefficients, A_i, is derived and written in Equation (3.42). Here, n is the degree of the polynomial and r is the coefficient subscript.

$$
A_r = \frac{1}{\Delta s^{n-r}} \sum_{i=0}^{r} \frac{(n-i)!}{(r-i)!(n-r)!} (-s_0)^{n-i} \alpha_i \qquad r \le n \tag{3.42}
$$

After obtaining the de-normalized feed correction polynomial, the mean squared error (MSE) between the true and predicted spline parameter values (i.e. u_i and $\hat{u}_i$) is calculated using Equation (3.43) and checked against a specified tolerance ε_{MSE}. If the tolerance is violated, then the (u_i, s_i) points are split into two sets of the same size and a spline is fit to each set with the aforementioned approach and checked against the MSE condition. N_Σ is the number of points used in the curve fitting process for each polynomial, which decreases as the data points are subdivided.

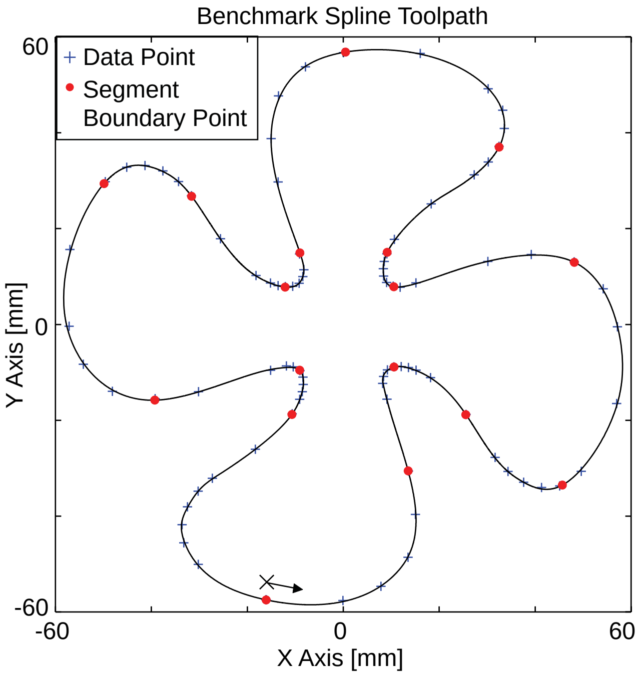

Figure 3-9. 17-segment NURBS toolpath

$$MSE = \sum_{i=1}^{N_\Sigma} \frac{(u_i - \hat{u}_i)^2}{N_\Sigma} < \varepsilon_{MSE} \qquad (3.43)$$

The division of the point data terminates when the feed correction polynomials satisfy Equation (3.43), or when the number of points in the fitting set is equal to the order of the polynomial. The resulting polynomials can then be embedded into the NC code for real-time feed correction during NURBS interpolation.

3.4 Simulation Results

To demonstrate the effectiveness of the NURBS toolpath parameterization and interpolation scheme, two example toolpaths are used. Toolpath 1 is generated from the fan-shaped spline data points, obtained from [9] with 150% scaling. Cubic NURBS segments (p = 3) with 6 control points (n = 5) are fitted to the data points with G^2 continuity including the closed loop constraints, as illustrated in Figure 3-9. The last segment has 9 control points. The data points and beta parameters used in the curve fitting and tangent and curvature continuity constraint equations are listed in Appendix A, along with the resulting knot vectors and control points from the toolpath parameterization.

The fan-shaped toolpath is interpolated at a sampling frequency of 1000 Hz with a jerk limited constant feedrate profile of 50 mm/s. Feed generation will be explained in Chapter 4. The arc-length of each segment is calculated with a tolerance of $\varepsilon = 1e-12$. The required number of feed corrections polynomials per segment, to achieve an MSE tolerance of $\varepsilon_{MSE} = 1e-10$, varies between five and twelve.

Interpolation with the feed correction polynomial is compared to natural interpolation in Figure 3-10 to show that the parameterization method generally doesn't produce arc-length parameterized curves and that reparameterization with the feed correction polynomial successfully minimizes unwanted feedrate fluctuations.

The resultant feedrate, $\hat{\dot{s}}_k$, is calculated with the numerical derivatives of the interpolated x-and y-axis position commands, $\hat{\dot{x}}$ and $\hat{\dot{y}}_k$, respectively, using Equation (3.44). T_s is the sampling period and *k* is the sample index between 2 and the number of samples, N_t minus one.

$$\hat{\dot{x}}_k = \frac{(x_{k+1} - x_{k-1})}{2T_s}, \quad \hat{\dot{y}}_k = \frac{(y_{k+1} - y_{k-1})}{2T_s} \quad \Rightarrow \hat{\dot{s}}_k = \sqrt{\hat{\dot{x}}_k{}^2 + \hat{\dot{y}}_k{}^2} \quad 2 \le k \le N_t - 1 \qquad (3.44)$$

The feed profile of the natural interpolation exhibits a maximum feed of 68.98 mm/s and a minimum of 33.79 mm/s during constant feedrate command. This translates to approximately 32 % to 37 % feed fluctuation. The maximum feed in the interpolated profile with feed correction is 50.046 mm/s, and the minimum is 49.942 mm/s. By applying the feed correction polynomial, maximum feed fluctuation is reduced to between 0.09 % and 0.11 %.

Acceleration and jerk profiles are also computed by numerical differentiation, similar to Equation (3.44), which reveals discontinuities that occur in the final axis position commands. The minimum and maximum acceleration and jerk values that occur in the interpolated profiles are listed in Table 3-2. Natural interpolation yields acceleration values that are approximately an order of magnitude higher than the interpolation with feed correction. Jerk values are two orders of magnitude greater.

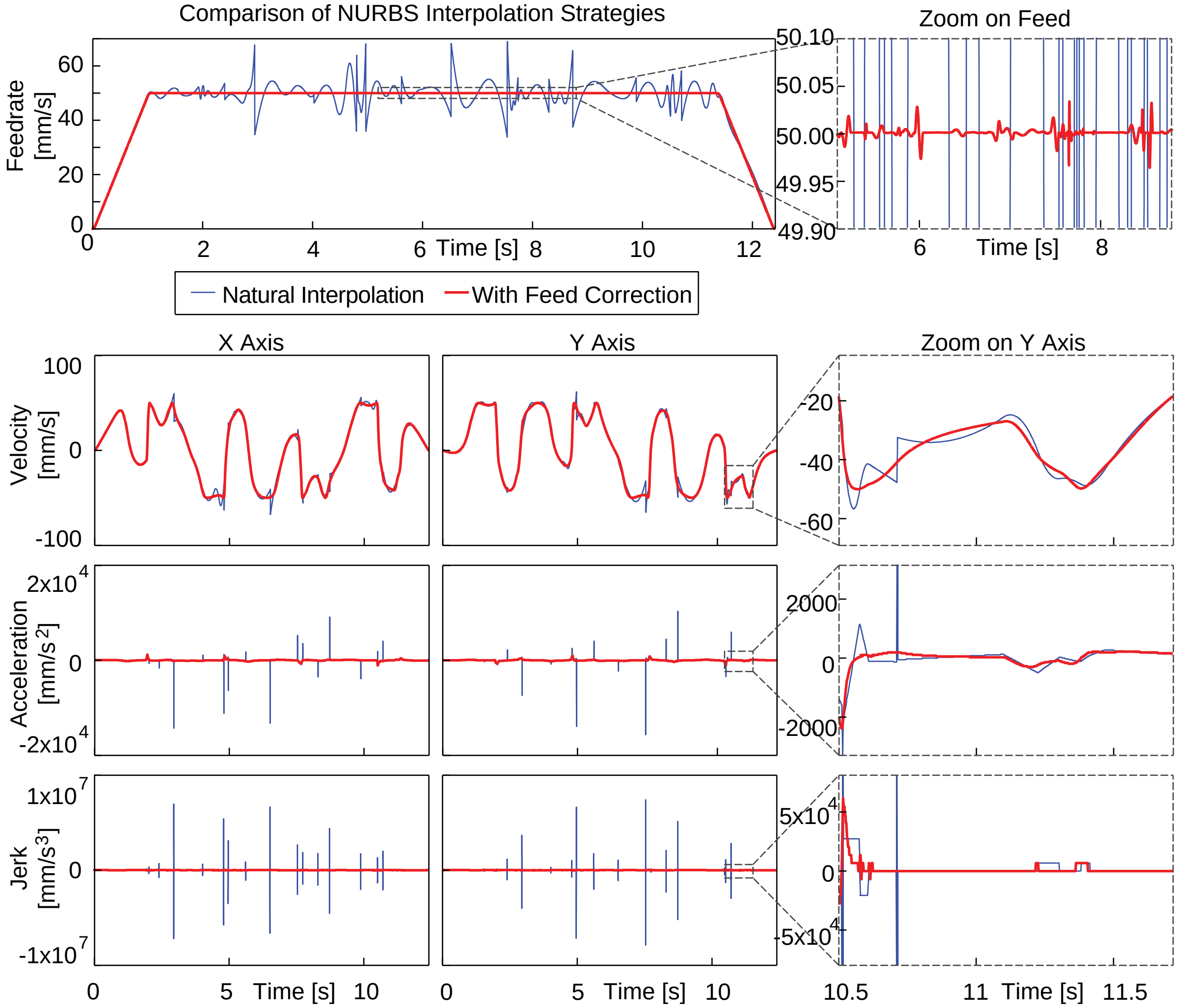

Figure 3-10. Toolpath 1 - comparison of NURBS interpolation without and with feed correction.

Table 3-2. Toolpath 1 - comparison of min/max axis acceleration and jerk values.

	Acceleration [mm/s^2]		Jerk [mm/s^3]	
	X Axis	Y Axis	X Axis	Y Axis
Natural Interpolation	-1.429 x 10^4 0.9083 x 10^4	-1.562 x 10^4 1.029 x 10^4	-7.194 x 10^6 6.946 x 10^6	-7.867 x 10^6 7.391 x 10^6
With Feed Correction	-0.1088 x 10^4 0.1202 x 10^4	-0.1199 x 10^4 0.1005 x 10^4	-5.742 x 10^4 3.602 x 10^4	-3.363 x 10^4 5.624 x 10^4

Toolpath 2 is generated from CAD data of a tibial-plateau (lower knee joint) implant model with 200% scaling and ball end tool offset compensation applied. The closed toolpath consists of 67 cubic NURBS segments (*p* = 3) with seven control points each (*n* = 6). The command feedrate is 20 mm/s. Figure 3-11 shows the NURBS toolpath as well as the resultant feedrate along the toolpath for interpolation with and without the feed correction polynomial. In comparing the feedrate profiles, it is clear that feed correction plays a significant role in ensuring that the motion along the toolpath is smooth.

The minimum feed encountered in the feed profile with feed correction is 19.32 mm/s. However, this is due to the fact that the feed is too high to track the given geometry, which is labeled as a sharp corner in the vicinity of $x = 10\,\text{mm}$ and $y = -25\,\text{mm}$ in Figure 3-11, and occurs during the time interval of 3.389 s to 3.395 s. Omitting this interval from the analysis, the minimum feed becomes 19.92 mm/s, which is a 0.41% decrease from the desired command feed, which is 20 mm/s. The maximum feed of the interpolated profile with feed correction is 20.066 mm/s, which is only a 0.33 % feed fluctuation. On the other hand, the maximum value in the feed profile, interpolated with uniform parameter increments, is 28.43 mm/s, which is approximately 42% greater than the command feed. Moreover, the minimum feed that was encountered is 16.45 mm/s, which is 18 % less.

A comparison of the minimum and maximum acceleration and jerk values is also provided in Table 3-3. The feed correction reduces the worst-case acceleration values by approximately half. Consequently, the jerk magnitudes are an order of magnitude lower, compared to interpolation without the feed correction polynomial.

Table 3-3. Toolpath 2 - comparison of min/max axis acceleration and jerk values.

	Acceleration [mm/s^2]		Jerk [mm/s^3]	
	X Axis	Y Axis	X Axis	Y Axis
Natural Interpolation	-3.022 x 10^3 2.716 x 10^3	-4.105 x 10^3 4.285 x 10^3	-1.518 x 10^6 1.592 x 10^6	-2.167 x 10^6 2.273 x 10^6
With Feed Correction	-1.420 x 10^3 0.9491 x 10^3	-1.152 x 10^3 1.455 x 10^3	-0.7955 x 10^6 0.1866 x 10^6	-0.2808 x 10^6 0.7361 x 10^6

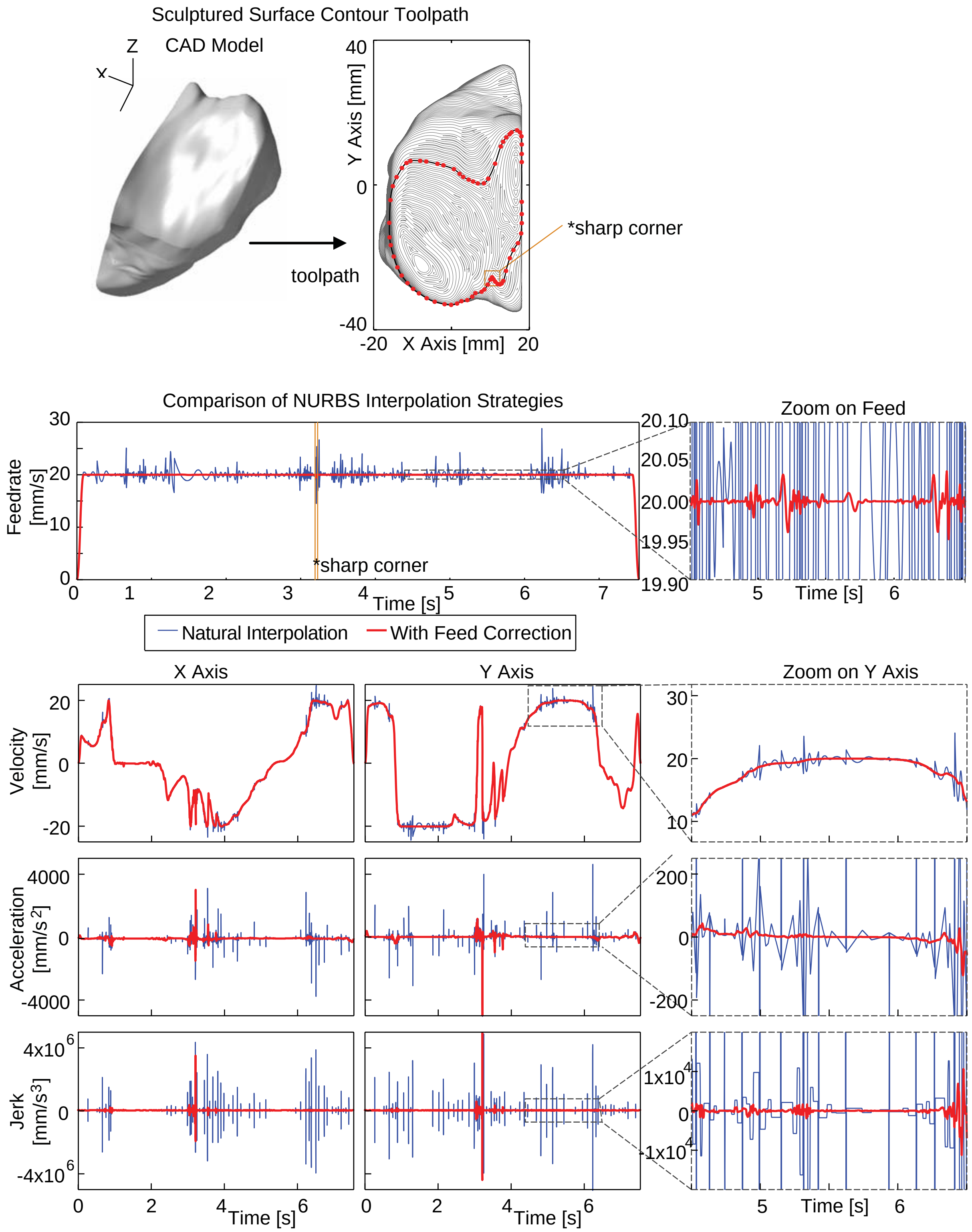

Figure 3-11. Toolpath 2 - comparison of resultant feedrate and axis kinematic profiles without and with feed correction.

3.5 Conclusions

This chapter has provided a basic mathematical framework for NURBS toolpath parameterization and interpolation. It has presented two problem formulations and solutions which utilize Lagrange Multipliers for solving two constrained curve fitting problems - the first to solve for NURBS control points, and the latter to obtain the coefficients of feed correction polynomials. The toolpath parameterization method guarantees that position G^0, tangent G^1, and curvature G^2 continuity is preserved at segment boundaries by utilizing beta-constraints. However, in general the toolpath parameterization presented here does not generate arc-length parameterized curves, which results in large feedrate fluctuations if the splines are interpolated with uniform parameter increments. In order to address this problem, the NURBS toolpaths are reparameterized with respect to arc-length with the feed correction polynomial prior to interpolation. This strategy minimizes unwanted feedrate fluctuations, regardless of the parameterization of the NURBS segment. The resultant feed profile of the interpolated trajectory with feed correction shows a significant reduction in feedrate fluctuations compared to the feed profile where feed correction was not applied.

One major advantage of spline toolpaths over conventional linear and circular toolpaths is the achievable continuity between segments which enables smooth continuous motion throughout the toolpath without having to come to a complete stop between segments. Rather, the feedrate can be continuously increased or decreased as deemed necessary by the toolpath geometry and dynamics of the machine tool, without incurring large acceleration or jerk in the axis feed drives.

The practicality and effectiveness of the proposed interpolation scheme has been demonstrated in simulation results. Experimental results will be presented in the following chapters. In Chapter 4, a generalized framework for continuous feedrate modulation is presented, which will be followed by the incorporation of a heuristic feed optimization strategy for overall NURBS trajectory generation in Chapter 5.

Chapter 4

Jerk Limited Feedrate Modulation

4.1 Introduction

In this chapter, a continuous feedrate modulation strategy for an *N*-segment toolpath is presented. Here, the proposed strategy seamlessly stitches multiple S-curve type feed transitions together as shown in Figure 4-1, to describe smooth motion along the toolpath. The strategy assumes that the toolpath is comprised of two or more segments. Jerk limited feed profiling for a single segment was presented by Erkorkmaz and Altintas [16].

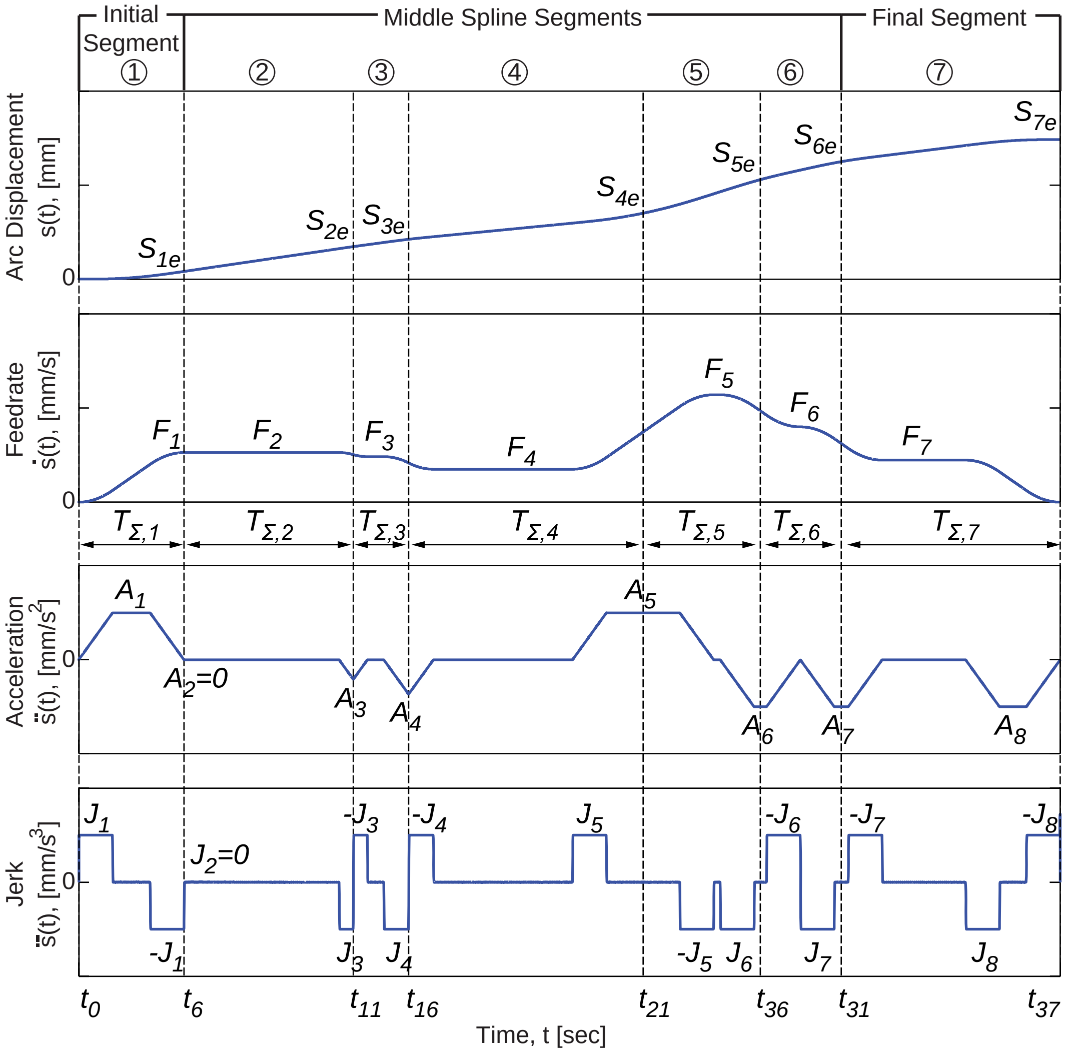

Figure 4-1. Feed modulation demonstrated with a 7-segment spline toolpath example.

Considering that the jerk profile is comprised of piecewise constant values, the acceleration profile is composed of linear and constant (zero and non-zero) functions which exhibit a characteristic trapezoidal profile. When the acceleration is zero, feed is held constant and the displacement increases linearly. When the acceleration is constant at a non-zero value, the feedrate is either linearly increasing or decreasing, and the displacement function is parabolic. When the acceleration function is linear, with a slope prescribed by the jerk value, the feedrate is then parabolic and the displacement is cubic.

The formulation of the kinematic equations is presented in Section 4.2. To ensure smooth transitions between piecewise functions within a segment and across segment boundaries, kinematic compatibility conditions are derived and presented in Section 4.3. In Section 4.4, implementation details of the feed modulation strategy are discussed. Experimental results are illustrated in Section 4.5 and conclusions are presented in Section 4.6.

4.2 Feed Profile Formulation

The ability to modify the feed on the fly allows the machining process to slow down for high curvature segments whilst maintaining high speeds throughout the rest of the toolpath. Here, the initial (*1*st), middle (*k*th), and final (*N*th) segment profiles are defined in order to establish a generalized framework for N segments, as shown in Figure 4-2, Figure 4-3, and Figure 4-4. The initial segment is defined at the start of the toolpath where the initial feedrate equals zero. As shown in Figure 4-2, a full acceleration transient is required to achieve the desired feedrate (F_1), which is realized in sub-segments ①, ②, and ③. Throughout the feed motion, the acceleration transients from the desired feed of one segment to that of the next are evenly distributed between consecutive segments. Therefore the initial segment also consists of a second partial acceleration transient that achieves half of the feed transition to the next segment's feed, in sub-segments ⑤ and ⑥. This results in the boundary feed to be the average of the desired feed values between two consecutive segments, as is the case for the final feed value $f_e = (F_1 + F_2)/2$ at the end of the *1*st feed segment. This mathematical relationship will be verified in the following section.

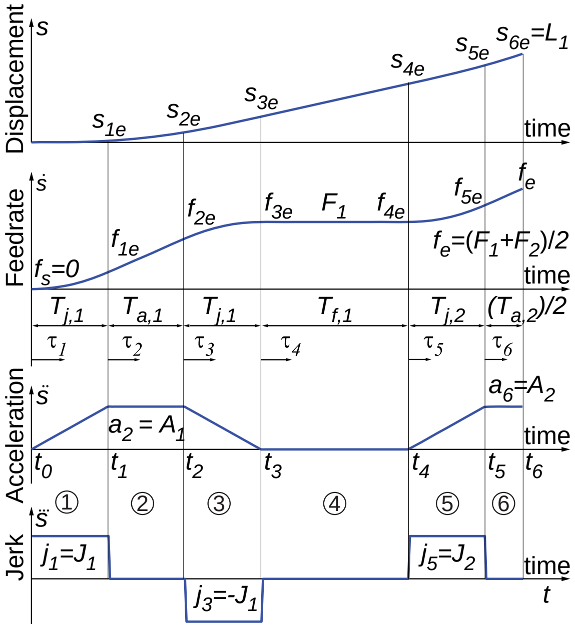

Figure 4-2. Initial (*1*st) segment profile.

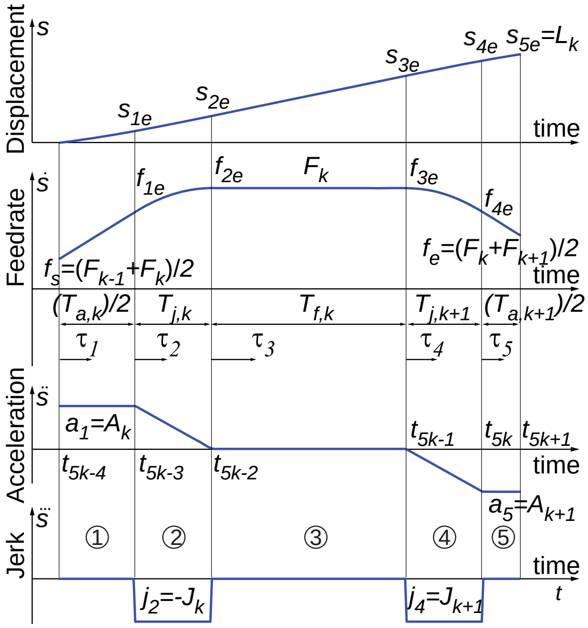

Figure 4-3. Middle (*k*th) segment profile.

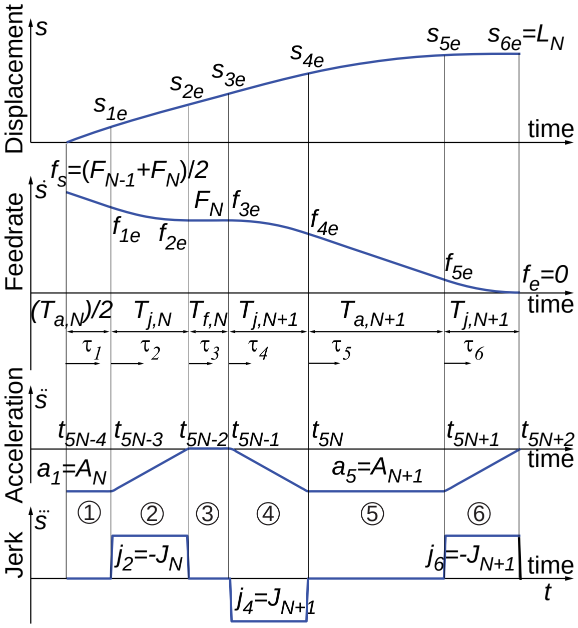

Figure 4-4. Final (*N*th) segment profile.

A middle segment, which is illustrated in Figure 4-3, is defined by non-zero start and end feeds, and consists of two partial acceleration transients: the end of the feed transition from the previous segment, and the start of the feed transition into the next one. The final segment, shown in Figure 4-4, concludes the motion along the toolpath with a final feed (i.e. tangential velocity) value of zero. Its first acceleration transient is partial (sub-segments ①-②), while its second acceleration transient is a full trapezoid that brings the motion to a full stop (sub-segments ④-⑥). Based on the illustrated piecewise constant jerk profiles, the acceleration, feed, and displacement profiles can be obtained by performing integration with respect to time.

Mathematically, given the initial conditions at time t_i $(i = 0,1,\ldots,5N+1)$, the tangential acceleration $\ddot{s}(t)$, feedrate $\dot{s}(t)$, and displacement $s(t)$ profiles can be obtained by integrating the tangential jerk profile $\dddot{s}(t)$ as follows:

$$\ddot{s}(t) = \ddot{s}(t_i) + \int_{t_i}^{t} \dddot{s}(\tau)d\tau, \quad \dot{s}(t) = \dot{s}(t_i) + \int_{t_i}^{t} \ddot{s}(\tau)d\tau, \quad s(t) = s(t_i) + \int_{t_i}^{t} \dot{s}(\tau)d\tau \tag{4.1}$$

Above, τ is the integration variable that represents time. In the following, kinematic equations are formulated with piecewise functions for the initial (*1*st), middle (*k*th), and final (Nth) segments.

The jerk profiles in Figure 4-2, Figure 4-3, and Figure 4-4 can be written as follows:

$$\dddot{s}_1(t) = \begin{cases} J_1, & 0 \le t < t_1 \\ 0, & t_1 \le t < t_2 \\ -J_1, & t_2 \le t < t_3 \\ 0, & t_3 \le t < t_4 \\ J_2, & t_4 \le t < t_5 \\ 0, & t_5 \le t < t_6 \end{cases} \tag{4.2}$$

$$\dddot{s}_k(t) = \begin{cases} 0, & t_{5k-4} \le t < t_{5k-3} \\ -J_k, & t_{5k-3} \le t < t_{5k-2} \\ 0, & t_{5k-2} \le t < t_{5k-1} \\ J_{k+1}, & t_{5k-1} \le t < t_{5k} \\ 0, & t_{5k} \le t < t_{5k+1} \end{cases} \tag{4.3}$$

$$\dddot{s}_N(t)=\begin{cases}0, & t_{5N-4}\le t<t_{5N-3}\\ -J_N, & t_{5N-3}\le t<t_{5N-2}\\ 0, & t_{5N-2}\le t<t_{5N-1}\\ J_{N+1}, & t_{5N-1}\le t<t_{5N}\\ 0, & t_{5N}\le t<t_{5N+1}\\ -J_{N+1}, & t_{5N+1}\le t\le t_{5N+2}\end{cases} \tag{4.4}$$

Above, t denotes absolute time and $t_1,t_2,\ldots,t_{5N+2}$ are time boundaries for each phase (i.e. sub-segment) in the entire profile. The initial and final segments have six phases of motion while the mid-segments only have five. Each segment has two acceleration transients in its motion profile that are either full or half trapezoids. J_k and J_{k+1} are the jerk values for the first and second acceleration transients, respectively, in spline segment k for $k=1,2,\ldots,N$. If the acceleration transient produces a change in feedrate to a higher speed, then J_k is positive. Otherwise, J_k is negative.

Integrating Equations (4.2)-(4.4) with respect to time, the acceleration profiles can be obtained as shown in Equations (4.5)-(4.7). The constant acceleration values for the first and second acceleration transients of the *k*th segment are denoted as A_k and A_{k+1} respectively, Here, τ_d is a relative time parameter that starts at the beginning of the *d*th phase, where $d=1,2,\ldots,6$ for the first and *N*th segments, while $d=1,2,\ldots,5$ for all middle segments ($2\le k\le N-1$).

$$\ddot{s}_1(t)=\begin{cases}J_1\tau_1, & 0\le t<t_1\\ A_1, & t_1\le t<t_2\\ A_1-J_1\tau_3, & t_2\le t<t_3\\ 0, & t_3\le t<t_4\\ J_2\tau_5, & t_4\le t<t_5\\ A_2, & t_5\le t<t_6\end{cases} \tag{4.5}$$

$$\ddot{s}_k(t) = \begin{cases} A_k, & t_{5k-4} \le t < t_{5k-3} \\ A_k - J_k \tau_2, & t_{5k-3} \le t < t_{5k-2} \\ 0, & t_{5k-2} \le t < t_{5k-1} \\ J_{k+1} \tau_4, & t_{5k-1} \le t < t_{5k} \\ A_{k+1}, & t_{5k} \le t < t_{5k+1} \end{cases} \tag{4.6}$$

$$\ddot{s}_N(t) = \begin{cases} A_N, & t_{5N-4} \le t < t_{5N-3} \\ A_N - J_N \tau_2, & t_{5N-3} \le t < t_{5N-2} \\ 0, & t_{5N-2} \le t < t_{5N-1} \\ J_{N+1} \tau_4, & t_{5N-1} \le t < t_{5N} \\ A_{N+1}, & t_{5N} \le t < t_{5N+1} \\ A_{N+1} - J_{N+1} \tau_6, & t_{5N+1} \le t \le t_{5N+2} \end{cases} \tag{4.7}$$

The feedrate profiles are obtained by integrating Equations (4.5)-(4.7) with respect to time as shown in Equations (4.8)-(4.10). f_s denotes the start feed, F_k is the desired feed of the *k*th segment to be achieved by the end of the first acceleration transient, and f_{de} is the feedrate reached at the end of the *d*th phase. $T_{j,k}$, $T_{a,k}$, and $T_{f,k}$ are the time durations of specific motion phases. The first subscript identifies the type of motion in the phase. For example, T_j refers to a non-zero jerk phase, T_a refers to a phase with constant non-zero acceleration, and T_f refers to the time duration of a constant feed phase. The second subscript identifies the segment to which the acceleration transient duration corresponds. For example, considering the *k*th segment, if the subscript is also *k*, this represents the *1*st acceleration transient. Otherwise, if the subscript is *k+1*, then this corresponds to the *2*nd acceleration transient.

$$\dot{s}_1(t) = \left\{ \begin{array}{lll} f_s + \dfrac{1}{2} J_1 \tau_1^2, & 0 \le t < t_1, & f_s = 0 \\ f_{1e} + A_1 \tau_2, & t_1 \le t < t_2, & f_{1e} = f_s + \dfrac{1}{2} J_1 T_{j,1}^2 \\ f_{2e} + A_1 \tau_3 - \dfrac{1}{2} J_1 \tau_3^2, & t_2 \le t < t_3, & f_{2e} = f_{1e} + A_1 T_{a,1} \\ f_{3e}, & t_3 \le t < t_4, & f_{3e} = f_{2e} + A_1 T_{j,1} - \dfrac{1}{2} J_1 T_{j,1}^2 \\ f_{4e} + \dfrac{1}{2} J_2 \tau_5^2, & t_4 \le t < t_5, & f_{4e} = f_{3e} \\ f_{5e} + A_2 \tau_6, & t_5 \le t < t_6, & f_{5e} = f_{4e} + \dfrac{1}{2} J_2 T_{j,2}^2 \end{array} \right. \tag{4.8}$$

$$\dot{s}_k(t) = \left\{ \begin{array}{lll} f_s + A_k \tau_1 & t_{5k-4} \le t < t_{5k-3}, & f_s = \left(F_{k-1} + F_k\right)/2 \\ f_{1e} + A_k \tau_2 - \dfrac{1}{2} J_k \tau_2^2 & t_{5k-3} \le t < t_{5k-2}, & f_{1e} = f_s + \dfrac{1}{2} A_k T_{a,k} \\ f_{2e} & t_{5k-2} \le t < t_{5k-1}, & f_{2e} = f_{1e} + A_k T_{j,k} - \dfrac{1}{2} J_k T_{j,k}^2 \\ f_{3e} + \dfrac{1}{2} J_{k+1} \tau_4^2 & t_{5k-1} \le t < t_{5k}, & f_{3e} = f_{2e} \\ f_{4e} + A_{k+1} \tau_5 & t_{5k} \le t < t_{5k+1}, & f_{4e} = f_{3e} + \dfrac{1}{2} J_{k+1} T_{j,k+1}^2 \end{array} \right. \tag{4.9}$$

$$\dot{s}_N(t) = \left\{ \begin{array}{ll} f_s + A_N \tau_1 & t_{5N-4} \le t < t_{5N-3}, \\ f_{1e} + A_N \tau_2 - \dfrac{1}{2} J_N \tau_2^2 & t_{5N-3} \le t < t_{5N-2}, \\ f_{2e} & t_{5N-2} \le t < t_{5N-1}, \\ f_{3e} + \dfrac{1}{2} J_{N+1} \tau_4^2 & t_{5N-1} \le t < t_{5N}, \\ f_{4e} + A_{N+1} \tau_5 & t_{5N} \le t < t_{5N+1}, \\ f_{5e} + A_{N+1} \tau_6 - \dfrac{1}{2} J_{N+1} \tau_6^2 & t_{5N+1} \le t \le t_{5N+2}, \end{array} \right.$$

$$\Rightarrow \left\{ \begin{array}{l} f_s = \left(F_{N-1} + F_N\right)/2 \\ f_{1e} = f_s + \dfrac{1}{2} A_N T_{a,N} \\ f_{2e} = f_{1e} + A_N T_{j,N} - \dfrac{1}{2} J_N T_{j,N}^2 \\ f_{3e} = f_{2e} \\ f_{4e} = f_{3e} + \dfrac{1}{2} J_{N+1} T_{j,N+1}^2 \\ f_{5e} = f_{4e} + A_{N+1} T_{a,N+1} \end{array} \right. \tag{4.10}$$

Finally, integrating Equations (4.8)-(4.10) with respect to time results in the following displacement profiles in Equations (4.11)-(4.13), where s_s is the start displacement, s_{de} is the displacement reached at the end of the dth phase, and $T_{f,k}$ is the time duration of the constant feed phase for the k th segment.

$$s_1(t) = \begin{cases} s_s + f_s\tau_1 + \frac{1}{6}J_1\tau_1^3 & t_0 \le t < t_1, \\ s_{1e} + f_{1e}\tau_2 + \frac{1}{2}A_1\tau_2^2 & t_1 \le t < t_2, \\ s_{2e} + f_{2e}\tau_3 + \frac{1}{2}A_1\tau_3^2 - \frac{1}{6}J_1\tau_3^3 & t_2 \le t < t_3, \\ s_{3e} + f_{3e}\tau_4 & t_3 \le t < t_4, \\ s_{4e} + f_{4e}\tau_5 + \frac{1}{6}J_2\tau_5^3 & t_4 \le t < t_5, \\ s_{5e} + f_{5e}\tau_6 + \frac{1}{2}A_2\tau_6^2 & t_5 \le t < t_6, \end{cases}$$

$$\Rightarrow \begin{cases} s_s = 0 \\ s_{1e} = s_s + f_sT_{j,1} + \frac{1}{6}J_1T_{j,1}^3 \\ s_{2e} = s_{1e} + f_{1e}T_{a,1} + \frac{1}{2}A_1T_{a,1}^2 \\ s_{3e} = s_{2e} + f_{2e}T_{j,1} + \frac{1}{2}A_1T_{j,1}^2 - \frac{1}{6}J_1T_{j,1}^3 \\ s_{4e} = s_{3e} + f_{3e}T_{f,1} \\ s_{5e} = s_{4e} + f_{4e}T_{j,2} + \frac{1}{6}J_2T_{j,2}^3 \end{cases} \qquad (4.11)$$

$$
\begin{gathered}
s_k(t) = \begin{cases}
s_s + f_s\tau_1 + \frac{1}{2}A_k\tau_1^2 & t_{5k-4} \le t < t_{5k-3} \\
s_{1e} + f_{1e}\tau_2 + \frac{1}{2}A_k\tau_2^2 - \frac{1}{6}J_k\tau_2^3 & t_{5k-3} \le t < t_{5k-2}, \\
s_{2e} + f_{2e}\tau_3 & t_{5k-2} \le t < t_{5k-1}, \\
s_{3e} + f_{3e}\tau_4 + \frac{1}{6}J_{k+1}\tau_4^3 & t_{5k-1} \le t < t_{5k}, \\
s_{4e} + f_{4e}\tau_5 + \frac{1}{2}A_{k+1}\tau_5^2 & t_{5k} \le t < t_{5k+1},
\end{cases} \\
\Rightarrow \begin{cases}
s_s = 0 \\
s_{1e} = s_s + \frac{1}{2}f_sT_{a,k} + \frac{1}{8}A_kT_{a,k}^2 \\
s_{2e} = s_{1e} + f_{1e}T_{j,k} + \frac{1}{2}A_kT_{j,k}^2 - \frac{1}{6}J_kT_{j,k}^3 \\
s_{3e} = s_{2e} + f_{2e}T_{f,k} \\
s_{4e} = s_{3e} + f_{3e}T_{j,k+1} + \frac{1}{6}J_{k+1}T_{j,k+1}^3
\end{cases}
\end{gathered}
\tag{4.12}
$$

$$
\begin{gathered}
s_N(\tau) = \begin{cases}
s_s + f_s\tau_1 + \frac{1}{2}A_N\tau_1^2, & t_{5N-4} \le t < t_{5N-3} \\
s_{1e} + f_{1e}\tau_2 + \frac{1}{2}A_N\tau_2^2 - \frac{1}{6}J_N\tau_2^3, & t_{5N-3} \le t < t_{5N-2}, \\
s_{2e} + f_{2e}\tau_3, & t_{5N-2} \le t < t_{5N-1}, \\
s_{3e} + f_{3e}\tau_4 + \frac{1}{6}J_{N+1}\tau_4^3, & t_{5N-1} \le t < t_{5N}, \\
s_{4e} + f_{4e}\tau_5 + \frac{1}{2}A_{N+1}\tau_5^2, & t_{5N} \le t < t_{5N+1}, \\
s_{5e} + f_{5e}\tau_6 + \frac{1}{2}A_{N+1}\tau_6^2 - \frac{1}{6}J_{N+1}\tau_6^3, & t_{5N+1} \le t \le t_{5N+2},
\end{cases} \\
\Rightarrow \begin{cases}
s_s = 0 \\
s_{1e} = s_s + \frac{1}{2}f_sT_{a,N} + \frac{1}{8}A_NT_{a,N}^2 \\
s_{2e} = s_{1e} + f_{1e}T_{j,N} + \frac{1}{2}A_NT_{j,N}^2 - \frac{1}{6}J_NT_{j,N}^3 \\
s_{3e} = s_{2e} + f_{2e}T_{f,N} \\
s_{4e} = s_{3e} + f_{3e}T_{j,N+1} + \frac{1}{6}J_{N+1}T_{j,N+1}^3 \\
s_{5e} = s_{4e} + f_{4e}T_{a,N+1} + \frac{1}{2}A_{N+1}T_{a,N+1}^2
\end{cases}
\end{gathered}
\tag{4.13}
$$

Using the formulated kinematic equations, it is possible to derive expressions for the time durations during each phase based on the given values of jerk J_k, acceleration A_k, feed F_k, and segment displacement length L_k. The expressions for $T_{j,k}$, $T_{a,k}$, and $T_{f,k}$ are summarized here. From the trapezoidal nature of the acceleration transient, the acceleration value A_k is equal to the area under the jerk block, which is equal to the jerk value J_k multiplied by the time duration of the jerk phase $T_{j,k}$. Hence, the time duration of the constant non-zero jerk phases can be written as follows:

$$T_{j,k} = \frac{A_k}{J_k} \tag{4.14}$$

The time duration of the constant non-zero acceleration phases, which are split in half where two adjacent segments connect (see Figure 4-1), can be obtained by integrating the area under the full trapezoidal acceleration transient. The area under the acceleration profile must be equal to the desired change in the feed profile. The feed at the start of the *k*th acceleration transient is F_{k-1} and the desired feed to be reached by the end of the acceleration transient is F_k. Using Equation (4.9), the feed at the end of phase ②, f_{2e}, is set to F_k. Hence, solving for the constant acceleration time duration yields the following expression:

$$T_{a,k} = \frac{F_k - F_{k-1}}{A_k} - \frac{A_k}{J_k} \tag{4.15}$$

Using Equations (4.14) and (4.15), it can be verified that the final feed value at the end of the *1*st segment is indeed $f_e = (F_1 + F_2)/2$. Using Equation (4.8), the feed at the end of phase ⑥ in Figure 4-2, can be expressed as:

$$f_e = F_1 + \frac{1}{2} A_2 (T_{j,2} + T_{a,2}) \tag{4.16}$$

Substituting in expressions for $T_{j,2}$ and $T_{a,2}$ yields:

$$f_e = F_1 + \frac{1}{2} A_2 \left(\frac{A_2}{J_2} + \frac{F_2 - F_1}{A_2} - \frac{A_2}{J_2} \right) \tag{4.17}$$

Simplifying Equation (4.17) gives $f_e = (F_1 + F_2)/2$. The same process can be done for the *k*th segment. In general, the final feed of the *k*th segment is $f_e = (F_k + F_{k+1})/2$.

Finally, the time durations of the constant non-zero feed phases for the initial (*1*st), middle (*k*th) and final (*N*th) segments are listed in Equations (4.18), (4.19), and (4.20). These equations are obtained by expressing the total travel distance in the *1*st, *k*th and *N*th segments using Equations (4.11), (4.12), (4.13) and setting them equal to the segment arc-lengths, L_1, L_k, and L_N, respectively. Equations (4.8), (4.9), (4.10) are used to substitute in expressions for the end feeds of each phase. Lastly, expressions for $T_{j,k}$ and $T_{a,k}$ from Equations (4.14) and (4.15), respectively, are substituted in to express the constant feed time duration in terms of given feedrates, accelerations, jerks and segment arc-lengths. In these equations, Δ_k is defined as $\Delta_k = F_k - F_{k-1}$.

$$T_{f,1} = \frac{L_1}{F_1} - \frac{1}{F_1}\left(\frac{\Delta_1^2}{2A_1} + \frac{\Delta_1 A_1}{2J_1} - \frac{3\Delta_2^2}{8A_2} + \frac{A_2^3}{24J_2^2} + \frac{F_1 A_2}{2J_2} + \frac{F_2 \Delta_2}{2A_2} \right) \tag{4.18}$$

$$T_{f,k} = \frac{L_k}{F_k} - \frac{1}{F_k}\left(\begin{array}{l} \frac{3\Delta_k^2}{8A_k} - \frac{A_k^3}{24J_k^2} + \frac{F_k A_k}{2J_k} + \frac{F_{k-1}\Delta_k}{2A_k} \\ -\frac{3\Delta_{k+1}^2}{8A_{k+1}} + \frac{A_{k+1}^3}{24J_{k+1}^2} + \frac{F_k A_{k+1}}{2J_{k+1}} + \frac{F_{k+1}\Delta_{k+1}}{2A_{k+1}} \end{array} \right) \tag{4.19}$$

$$T_{f,N} = \frac{L_N}{F_N} - \frac{1}{F_N}\left(\begin{array}{l} \frac{3\Delta_N^2}{8A_N} - \frac{A_N^3}{24J_N^2} + \frac{F_N A_N}{2J_N} + \frac{F_{N-1}\Delta_N}{2A_N} \\ -\frac{\Delta_{N+1}^2}{2A_{N+1}} - \frac{A_{N+1}\Delta_{N+1}}{2J_{N+1}} \end{array} \right) \tag{4.20}$$

Detailed mathematical derivations for the above expressions are provided in Appendix B. These derived expressions are used to assess the kinematic feasibility of the motion defined by the given jerk, acceleration, feedrate, and displacement values.

4.3 Kinematic Compatibility Conditions

Kinematic compatibility conditions are derived based on the analytical expressions for the displacement, feedrate, acceleration, and jerk profiles, provided in the previous section. In this section, the resulting conditions are summarized. The maximum tangential acceleration and jerk limits are set by the process designer and are based on the machine's acceleration capabilities. Kinematic compatibility is defined as sufficient travel length to change the feedrate within the specified acceleration and jerk limits of the machine. If kinematic compatibility is possible, then the displacement, feedrate and acceleration profiles will be continuous, and the jerk profile will be limited.

Given specified values for the control loop sampling period T_s, the desired segment feedrates F_k, where F_0 and F_{N+1} are the initial and final feeds of the whole toolpath, and the acceleration and jerk magnitude limits, $A_{\max}$ and $J_{\max}$ respectively, the acceleration values A_k are calculated. Then, feed transitions are checked against compatibility conditions, based on the segment travel length L_k. If a condition is not satisfied, the violating feedrate value is modified to yield a kinematically compatible profile. The specified maximum jerk should not be larger than that which is achievable within the sampling period, given a maximum acceleration magnitude. Hence, the sign and magnitude of the jerk J_k is calculated as:

$$J_k = \mathrm{sgn}(F_k - F_{k-1}) \cdot \min(J_{\max}, A_{\max} / T_s) \qquad (4.21)$$

To achieve smooth feed transients, the appropriate acceleration magnitudes must be determined. The feed reached at the end of the first acceleration transient must equal the desired feed F_k. The maximum allowable acceleration to transition from F_{k-1} to F_k, assuming no constant acceleration phase, is found by setting the constant acceleration duration ($T_{a,k}$) to zero in Equation (4.15). Capped by a specified maximum acceleration magnitude $A_{\max}$, the acceleration A_k is determined with the following equation:

$$A_k = \mathrm{sgn}(F_k - F_{k-1}) \cdot \min\left(A_{\max}, \sqrt{(F_k - F_{k-1})J_k}\right) \qquad (4.22)$$

The feed compatibility condition determines whether there is sufficient travel length in a segment to carry out the desired motion specified by the feedrates of three consecutive segments, F_{k-1}, F_k, and F_{k+1}, their corresponding transition acceleration values, A_k, A_{k+1}, and jerk values J_k, J_{k+1}. The distance available for constant feed motion is denoted as $L_{f,k}$ and can be found by subtracting the distance traveled during the first and second acceleration transients from the total segment length, L_k. $L_{f,k}$ must be greater than or equal to zero. The resulting feed compatibility conditions for the initial (*1*st), middle (*k*th), and final (*N*th) segments are summarized in Equation (4.23).

$$
\begin{aligned}
L_{f,1} &= L_1 - \left[\frac{\Delta_1^2}{2A_1} + \frac{A_1\Delta_1}{2J_1} - \frac{3\Delta_2^2}{8A_2} + \frac{A_2^3}{24J_2^2} + \frac{F_1A_2}{2J_2} + \frac{F_2\Delta_2}{2A_2} \right] \geq 0 \\
L_{f,k} &= L_k - \left[\begin{array}{l} \dfrac{3\Delta_k^2}{8A_k} - \dfrac{A_k^3}{24J_k^2} + \dfrac{F_kA_k}{2J_k} + \dfrac{F_{k-1}\Delta_k}{2A_k} \\ -\dfrac{3\Delta_{k+1}^2}{8A_{k+1}} + \dfrac{A_{k+1}^3}{24J_{k+1}^2} + \dfrac{F_kA_{k+1}}{2J_{k+1}} + \dfrac{F_{k+1}\Delta_{k+1}}{2A_{k+1}} \end{array} \right] \geq 0 \\
L_{f,N} &= L_N - \left[\begin{array}{l} \dfrac{3\Delta_N^2}{8A_N} - \dfrac{A_N^3}{24J_N^2} + \dfrac{F_NA_N}{2J_N} + \dfrac{F_{N-1}\Delta_N}{2A_N} \\ -\dfrac{\Delta_{N+1}^2}{2A_{N+1}} - \dfrac{A_{N+1}\Delta_{N+1}}{2J_{N+1}} \end{array} \right] \geq 0
\end{aligned}
\tag{4.23}
$$

As defined earlier, $\Delta_k = F_k - F_{k-1}$.

4.4 Implementation Details

As feed transitions are spread across segment boundaries, kinematic compatibility in a segment is affected by its adjacent segments. In order to ascertain whether kinematic compatibility is satisfied across the segment boundaries, the previous and next segments also need to be tested against the feed compatibility conditions stated in Equation (4.23). Smooth and continuous motion is achievable if all three consecutive segments satisfy these conditions. A forward traversal through the toolpath checks the compatibility of the desired feedrates, which can be provided by the NC programmer, or obtained through a feed optimization routine such as the one that is presented in Chapter 5. Under certain situations, a

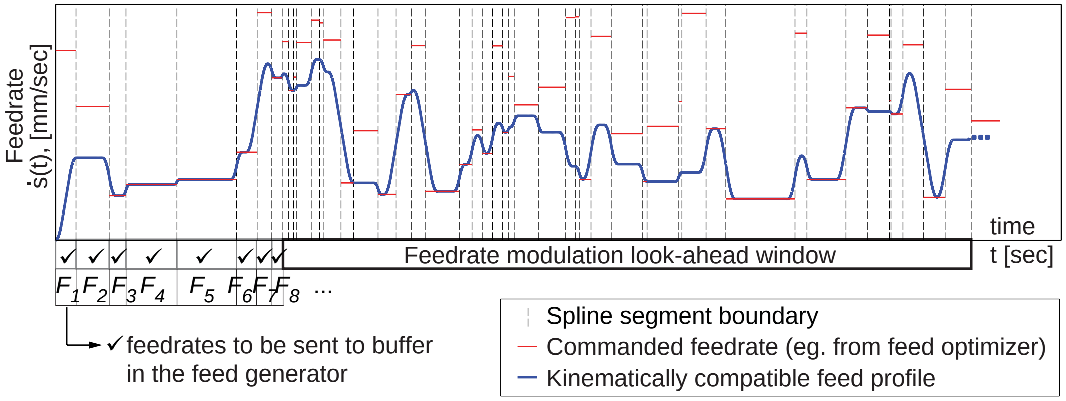

Figure 4-5. Implementation of feed modulation strategy with a look-ahead window.

kinematically compatible solution may not be achievable by simply modulating the commanded feed, acceleration, or jerk values. In such circumstances, the real-time interpolator needs to back-track through the planned feed values and perform adjustment to the earlier NC blocks, in order to yield a kinematically compatible feed profile for the current trajectory segment. Hence, a look-ahead buffer is implemented for this purpose as shown in Figure 4-5.

When the desired feedrate of a segment provided by the NC programmer or an optimization routine is incompatible, a bisection search method finds a kinematically feasible feed efficiently. The feeds of the previous and next segment are either fixed or free, where free means that it is set equal to the feed of the current segment. In forward planning, the feed of the previous segment is fixed and the feed of the next segment is generally free. The exception is when the desired feed of the next segment is lower than the test feed f_{mid}, in which case the feed of the next segment is fixed at its desired level, F_{k+1}. Similar rules apply for backward planning in reverse. The range of the search space is bounded by zero at the bottom, and the desired feedrate F_k at the top. The search algorithm bisects the feed search space iteratively, as shown in Figure 4-6, until a feasible solution is found within a specified tolerance F_{tol}. The number of iterations required to find the new compatible feed within the specified tolerance can be found with Equation (4.24).

$$n = round[\log_2(F_k / F_{tol})] + 1 \tag{4.24}$$

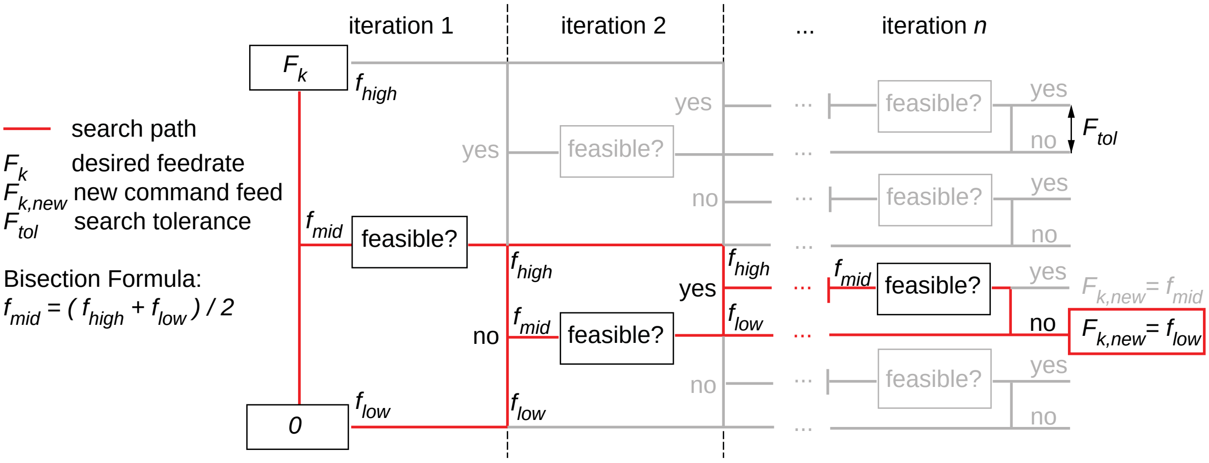

Figure 4-6. Bisection search algorithm for a feasible feed.

At each iteration, the test feed value is the bisected value, f_{mid}, of the search space defined by a low bound f_{low} and high bound f_{high}. f_{mid} is tested against the kinematic compatibility conditions in Equation (4.23). If there is a violation, then f_{mid} becomes the high bound of the next iteration's search space and the previous low bound remains the same. Otherwise, f_{mid} becomes the new low bound. On the last iteration, if the last tested feed is feasible, then it becomes the new command feedrate, $F_{k,new}$. On the other hand, if it is not feasible, then the new feedrate is the last feasible feed found in previous iterations. It is assumed that the search tolerance is smaller than the smallest feasible feed such that the new command feedrate is always greater than zero.

The kinematically compatible feeds are passed through the feed generator to obtain arc-length position commands at each control sample. These can be subsequently transferred to the real-time interpolator to generate individual axis motion commands using the NURBS interpolation strategy explained in Chapter 3.

4.4.1 Look-ahead for Long Toolpaths

A look-ahead window is sufficiently long if there is enough travel length to decelerate from the largest possible feed to zero, by using a series of trapezoidal acceleration transients as defined in Sections 4.2 and 4.3. An estimate for the number of segments that may be required can be obtained, given a maximum feed $F_{\max}$, acceleration and jerk magnitudes,

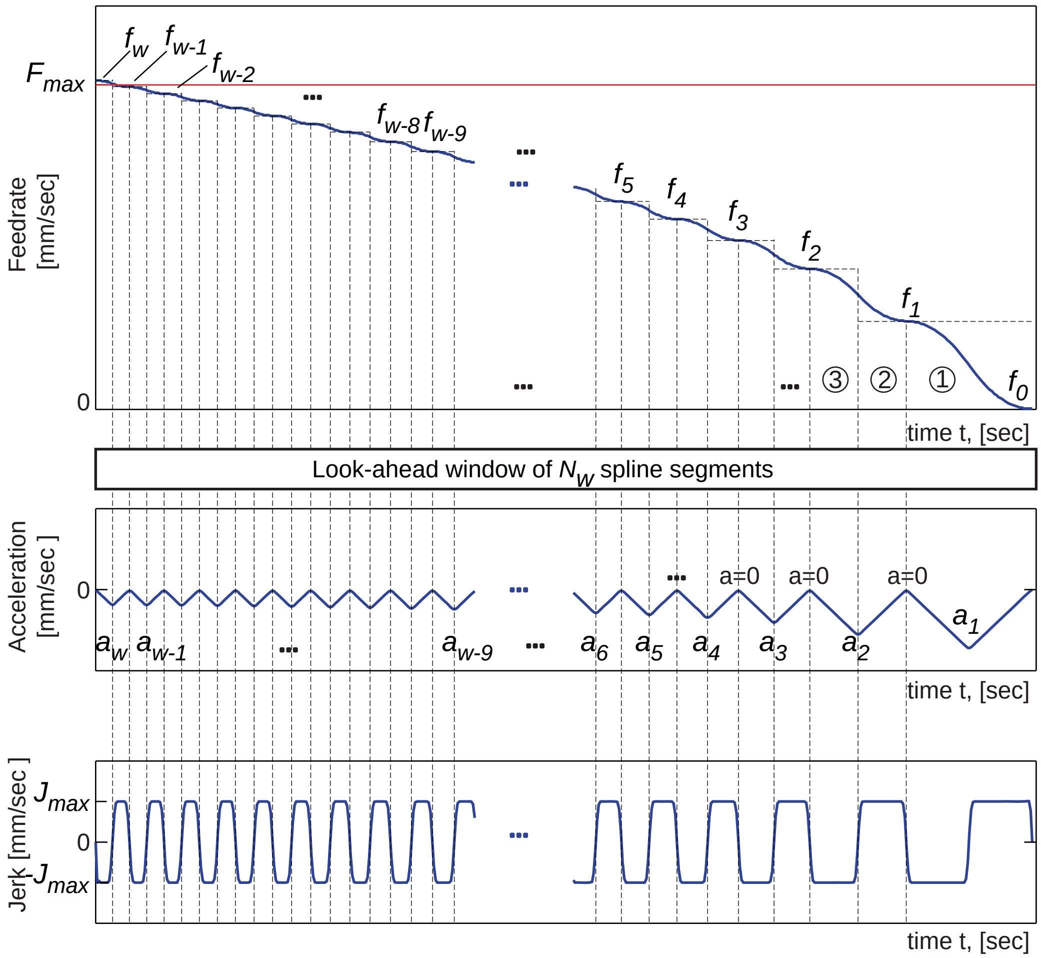

Figure 4-7. Deceleration from maximum feed to rest.

and by assuming that each segment has a specified path length that is larger than a minimum value, $L_{\min}$. The feed modulation framework facilitates a large deceleration in S-curve steps. Maximal feed change is realized by eliminating every other acceleration transient such that maximum acceleration occurs at every other segment boundary and zero acceleration occurs otherwise. An illustration of the deceleration process from $F_{\max}$ to zero is provided in Figure 4-7.

The resulting sequence of acceleration values is $\{a_w, 0, a_{w-1}, 0, \ldots, a_3, 0, a_2, 0, a_1\}$, and the command feedrate sequence is $\{f_w, f_{w-1}, f_{w-1}, \ldots, f_2, f_2, f_1, f_1, f_0\}$. For example, a_w is the acceleration transition value to decelerate from f_w to f_{w-1}. The total number of segments in the look-ahead window is denoted as N_w and defined as $N_w = 2w - 1$.

In order to estimate the number of required segments for the look-ahead window, the feed steps in the deceleration profile shown in Figure 4-7 are computed iteratively until a value that is greater than the maximum feedrate is reached. Essentially the deceleration profile is calculated in reverse, starting at the right-most segment in the figure, which is denoted as segment ①. To start, a counter variable w is initialized to zero and f_w is initialized to zero. The maximum allowable feed step, denoted as $\delta_w = f_w - f_{w-1}$, is calculated. By considering the first three phases of the initial segment type illustrated in Figure 4-2, Equations (4.8) and (4.11) can be reduced to calculate the maximum reachable feed within a given path length constraint. Considering that the time duration of the constant acceleration phase is zero, the travel length by the end of the third phase (s_{3e}) can be expressed in terms of the initial feed value f_{w-1}, the acceleration transition value a_w and time duration of the jerk phase, T_j.

$$s_{3e} = f_{w-1}T_j + a_w T_j^2 \tag{4.25}$$

Given a maximum jerk value $J_{\max}$ and substituting $T_j = a_w / J_{\max}$, from Equation (4.14), into the above equation, yields a cubic equation in a_w:

$$0 = a_w^3 + f_{w-1}J_{\max}a_w - s_{3e}J_{\max}^2 \tag{4.26}$$

From the above equation, a_w can be solved using the Newton-Raphson method with $A_{\max}$ as an initial guess. If $w = 1$, then s_{3e} is set to $L_{\min}$. Otherwise, s_{3e} is $L_{\min}$ multiplied by two (i.e. $2L_{\min}$) to reflect that the acceleration transient occurs over two segment lengths. Then, rearranging Equation (4.15) as $f_w - f_{w-1} = a_w T_a + a_w^2 J_{\max}$ with T_a set to zero, the maximum feed step δ_w in terms of the computed acceleration value and maximum jerk value can be obtained with the following equation:

$$\delta_w = \frac{a_w^2}{J_{\max}} \tag{4.27}$$

Finally, if the feed reached is greater than the maximum specified feed, that is if $f_{w-1} + \delta_w \geq F_{\max}$, then the value of the counter variable is used to calculate the window size, which is $N_w = 2w - 1$. Otherwise, the counter variable is incremented by one and the procedure to calculate the next feed step is repeated with Equations (4.25), (4.26), and (4.27).

The number of segments required to decelerate from a maximum feed of 150 mm/s within acceleration and jerk limits of 500 mm/s^2 and 10,000 mm/s^3, respectively, assuming a minimum travel length in each segment of 0.1 mm, is 22,501 segments.

4.5 Experimental Results

To demonstrate the effectiveness of the NURBS interpolation and continuous feed modulation strategy, surface machining tests were performed on a 3-axis router experimental setup illustrated in Figure 4-8. Drive parameter identification tests [39] were performed on the router to obtain control signal equivalent inertia, viscous damping, and Coulomb friction values for all three axes, which are listed in Table 4-1. The gantry design results in two axes in the x-direction which are labeled "Right" and "Left". Separate parameters were identified by assuming an independent relationship between the two axes, despite the fact that the identification data of the X axes were obtained simultaneously. The left X-axis appears to have lower control signal equivalent inertia and damping, and higher Coulomb friction compared to the right X-axis. The reason for this dissimilarity is perhaps due to the weight of the Z-axis which tended to operate closer to the right side of the gantry, simply because it is closer to the home position of (0,0,0). Thus, the perceived inertia would be greater on the right side rather than on the left.

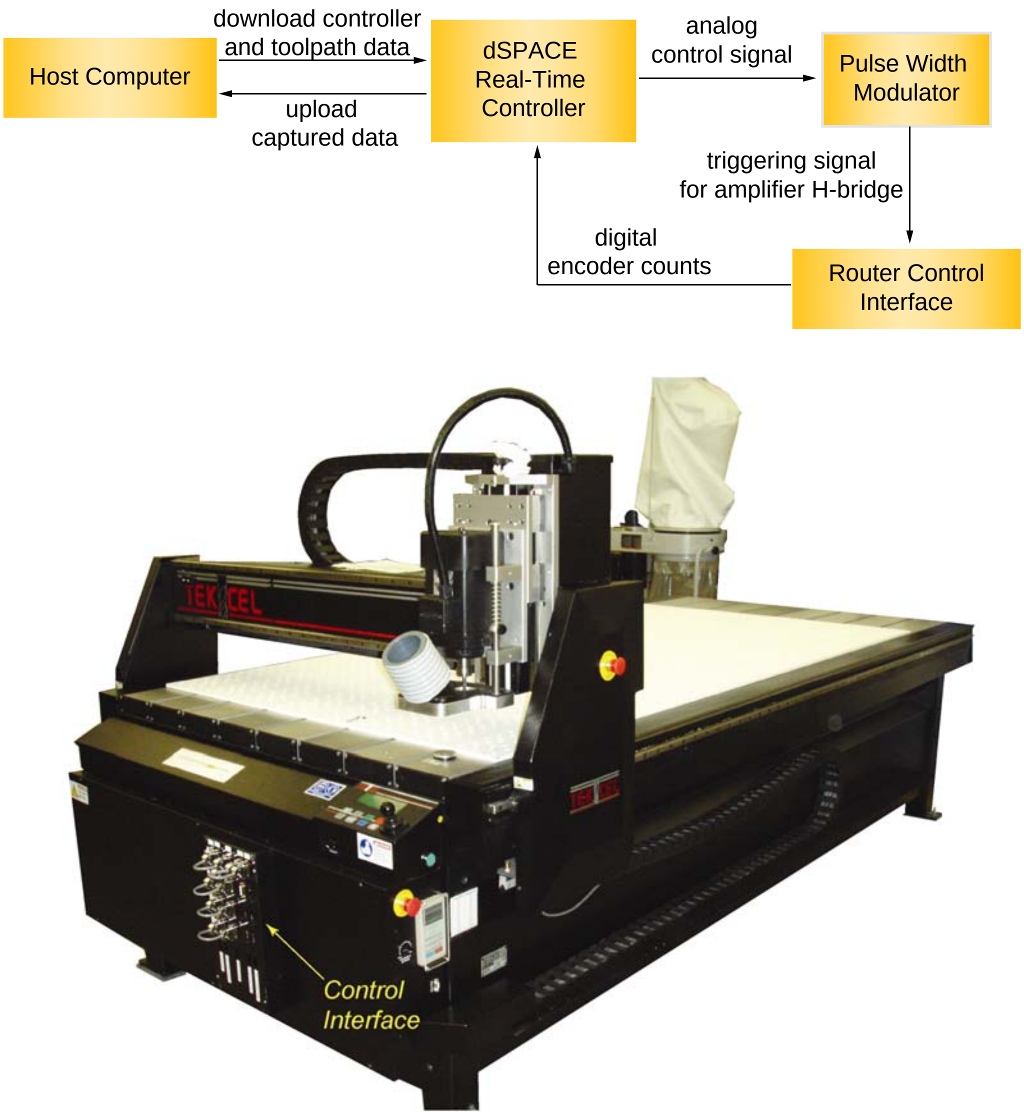

Figure 4-8. Experimental setup (4' x 8' router table).

Table 4-1. Identified control signal equivalent parameters of the experimental setup.

Axis	Inertia, m [V/(m/s^2)]	Damping, b [V/(m/s)]	Coulomb Friction, d_{coul} [V]	
			Positive	Negative
X (right)	2.0409	39.9446	0.3270	-0.3206
X (left)	1.6060	38.3927	0.4599	-0.4356
Y	1.0803	30.7299	1.9078e-004	-3.3471e-004
Z	2.5810	83.8100	0.2172	-0.2031

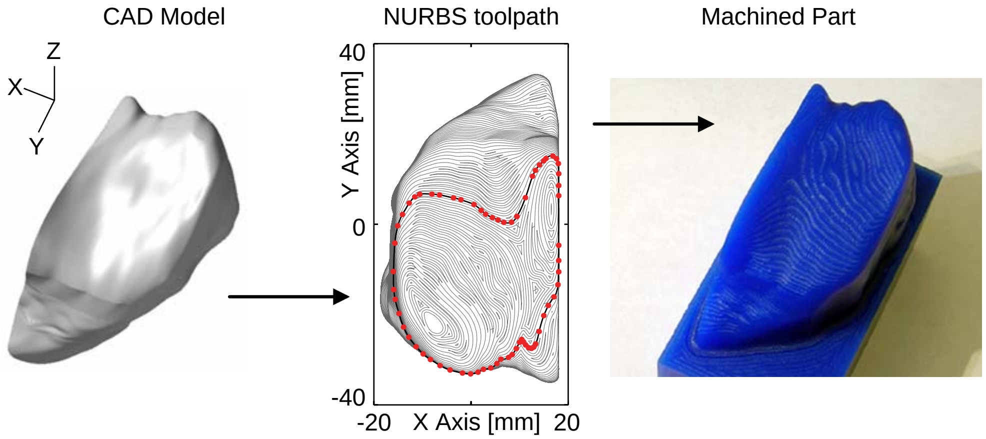

Figure 4-9. Biomedical implant model to machined part.

The controller design was implemented in MATLAB Simulink, and subsequently downloaded to the real-time dSPACE controller, as shown in Figure 4-8. The router is controlled with adaptive sliding mode control [40] at a bandwidth of 16 Hz. The tuned sliding mode control parameters are listed in Table 4-2. The dSPACE controller sends control signals to the router's driver board through a control interface after a pulse-width modulator converts the analog signals to digital signals. The dSPACE captures encoder counts directly from the motors' rotary encoders to close the feedback loop. The encoder resolution for the X and Y axes is 1/384 mm (2.6 um) and for the Z axis is 1/960 mm (1.04 um). The loop closure (interpolation) period was 1 ms. Also, to counteract the force of gravity, a constant 1 V signal is applied to the Z-axis.

Table 4-2. Tuned sliding mode control (SMC) parameters for the experimental setup.

Axis	λ [rad/s]	K_s [V/(rad/s^2)]	ρ [V/(rad/s)]
X (right)	100	1	100
X (left)	100	1	100
Y	100	1	100
Z	100	1	80

A tibial-plateau (lower knee joint) implant model was machined out of wax with 200% scaling as shown in Figure 4-9. MasterCAM was initially used to generate the tool center points (TCPs), which define linear toolpath segments that are within the specified machining tolerance of 25 um. A contour machining strategy was employed, which means that the toolpath consisted of X-Y contours at varying depths, for which the step down was specified as 0.5 mm. The NURBS toolpaths were generated from the CAD data in MasterCAM with ball end tool offset compensation, using the parameterization method presented in Chapter 3. A long length two-flute 3mm ball nose mill was used to machine the part. Given the axis velocity (150 mm/s), jerk (25,000 mm/s^3) and control signal (5 V) limits, selection of the commanded feed values was realized using the heuristic feed optimization technique presented in [37], which is the predecessor of the method presented in Chapter 5.

The feed motion profiles are shown in Figure 4-10. The velocity, acceleration and jerk profiles were calculated by taking the numerical derivatives of the position trajectory using Equation (4.28). Their smoothness indicates that the position trajectory was generated correctly without any unanticipated flaws or discontinuities. T_s is the sampling period, and k is the time step index, such that $t = kT_s$ for $1 \le k \le N_t - 1$.

$$\begin{gathered}\hat{v}_0 = \hat{a}_0 = \hat{j}_0 = 0, \qquad \hat{v}_{N_t} = \hat{a}_{N_t} = \hat{j}_{N_t} = 0, \\ \hat{v}_k = \frac{s_{k+1} - s_{k-1}}{2T_s}, \qquad \hat{a}_k = \frac{\hat{v}_{k+1} - \hat{v}_{k-1}}{2T_s}, \qquad \hat{j}_k = \frac{\hat{a}_{k+1} - \hat{a}_{k-1}}{2T_s}\end{gathered} \tag{4.28}$$

As can be seen in Figure 4-10, the kinematic profiles are smooth and limited in jerk in all axes, as originally planned. As a result the servo errors, defined as the difference between the reference position and the actual measured position (i.e. $x_{ref} - x_{meas}$), in the two axes do not exceed 15 um, which is only in the order of 6 encoder counts while operating the router at its top speeds and acceleration. Smooth feed motion ensures that the machine is able to track the given reference trajectories.

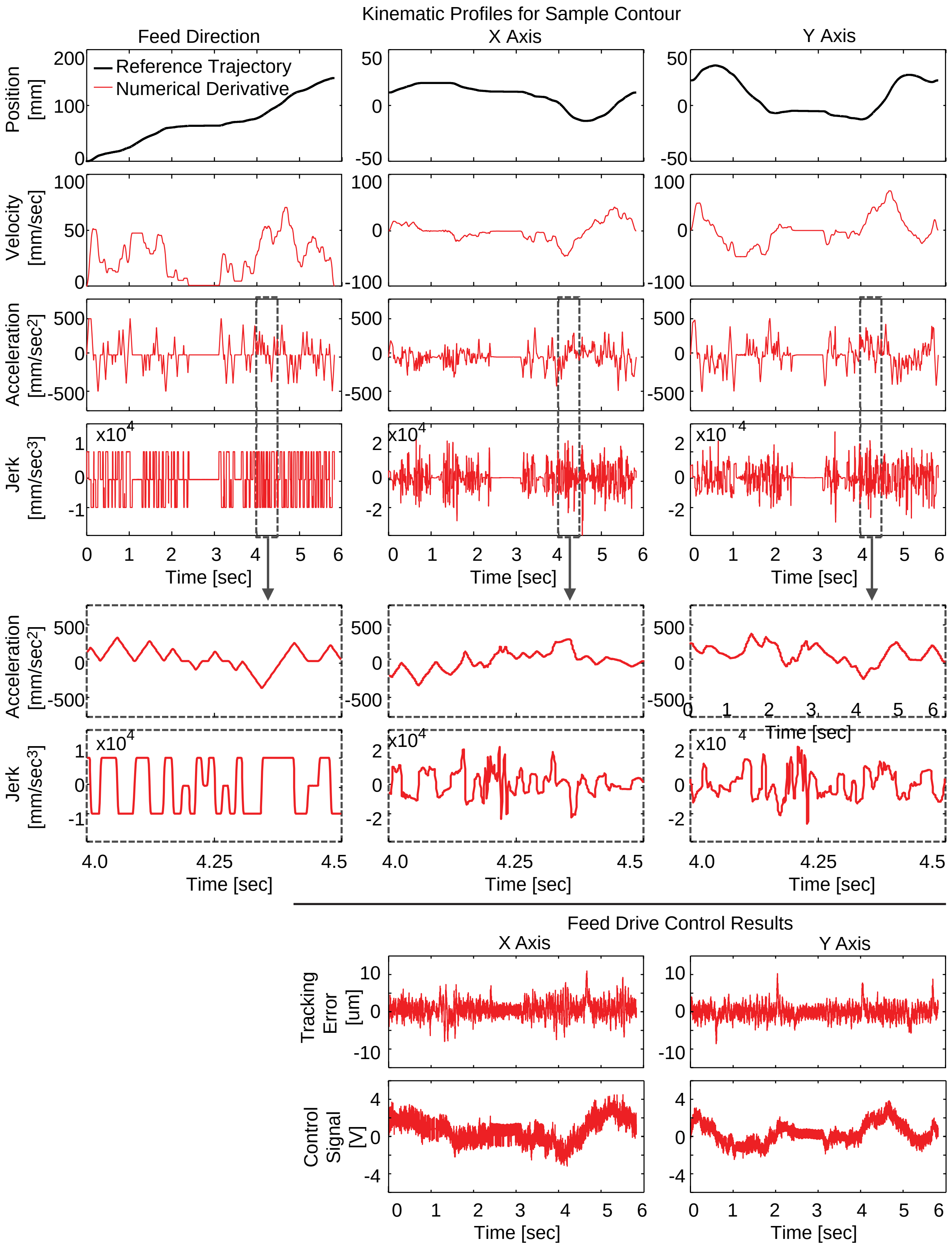

Figure 4-10. Kinematic profiles and controlled contouring results for the sample NURBS toolpath.

4.6 Conclusions

A generalized framework for feed modulation of an *N*-segment toolpath was developed and presented in this chapter. This framework is used to perform NURBS toolpath interpolation with continuous feedrate modulation. Smooth feed motion is ensured by the feedrate modulator, which utilizes analytically derived feed compatibility conditions to guarantee continuity in position, velocity, and acceleration profiles between neighboring segments. If the desired feedrate for a segment is kinematically infeasible, then a bisection search algorithm lowers the command feed to the highest feasible feedrate. Long toolpaths are handled by employing a look-ahead function in the feed modulator such that enough travel distance is available to bring the feed motion to a stop when needed. Hence, feedrate generation and trajectory interpolation are uninterrupted in the real-time CNC controller. The effectiveness of the proposed method has been demonstrated in machining a complex surface.

Chapter 5

Feedrate Optimization

5.1 Introduction

In this chapter, a computationally efficient feedrate optimization strategy is developed for spline toolpaths with jerk-limited feed profiling. The technique combines analytically derived compatibility equations from Section 4.3, with a heuristic search method, which helps generate feed profiles with reduced cycle time while adhering to axis velocity, acceleration, torque, and jerk constraints. The feed optimization is integrated with the feed modulation strategy presented in the previous chapter. Using the S-curve function allows the optimized feed profiles to be implemented on most existing CNC's. The proposed strategy yields shorter cycle time compared to the worst-case curvature approach [31], which is frequently used in industry, and converges faster than more elaborate gradient-based optimization techniques [32]. In the following, Section 5.2 presents the feedrate optimization problem and Section 5.3 presents the solution methodology of the newly developed optimization technique. The effectiveness of the new strategy is demonstrated in contour machining experiments in Section 5.4. The conclusions are summarized in Section 5.5.

5.2 Problem Formulation

The objective of feedrate optimization is to minimize the cycle time to machine a part, while preserving the desired contouring accuracy. In other words, the aim is to maximize the feedrate along the toolpath without compromising the quality of the final product. Considering an *N*-segment toolpath, where T_k represents the cycle time of the *k*th segment, the objective function can be expressed as a minimization of the total cycle time, as shown in Equation (5.1).

$$\min \sum_{k=1}^{N} T_k = \min(T_1 + T_2 + \cdots + T_N) \tag{5.1}$$

Conversely, the objective function can also be written as a maximization of the feedrate for the *k*th segment, F_k, where the feedrate of each segment is maximized individually as in Equation (5.2).

$$\max F_k , \quad 1 \le k \le N \tag{5.2}$$

The latter objective function means that if the tool is traveling at the maximum allowable feedrate for each segment, then the total cycle time is minimized.

The optimization constraints are chosen to ensure that the machine performs within the physical and control limits of its components and that the desired contouring accuracy during machining is maintained. For these reasons, constraints are imposed on the feedrate, and the velocities, motor torques, and jerks of all axes. Considering the machining process in general, the cutting forces are proportional to the feedrate. Excessive cutting forces are undesired as they can premature tool wear or breakage, which can either damage the part or the machine. Hence, the feedrate is limited by a maximum value to indirectly limit the resulting cutting forces. Naturally, the feedrate must also be greater than zero to avoid reverse motion along the toolpath. Thus,

$$0 < F_k \le F_{\max} \tag{5.3}$$

Axis velocity is constrained based on the physical limits of the axis drive. Ensuring that the drive doesn't exceed this limit also helps to prolong the life of the drive components, for example, the motor, ball screw and bearings. Thus, each axis velocity is bounded by a minimum and a maximum value, and compacted into matrix form as,

$$\left.\begin{array}{l} v_{x\min} \le v_x \le v_{x\max} \\ v_{y\min} \le v_y \le v_{y\max} \\ v_{z\min} \le v_z \le v_{z\max} \end{array}\right\} \Rightarrow \mathbf{v}_{\min} \le \mathbf{v} \le \mathbf{v}_{\max} \tag{5.4}$$

The demanded torque must not exceed the amount of torque that the motor can produce. Excessive torque demands results in saturation of the motor's actuators and excessive tracking error. The system may also become non-linear during actuator saturation and go into instability if there is integral action in the controller. In order to describe the torque demand, a dynamic model of the axis drives is required. In the case that a dynamic model is unavailable, process designers can impose limits on the commanded acceleration, which is correlated to the torque demand, such that:

$$\left.\begin{aligned} a_{x\,\min} \le a_x \le a_{x\,\max} \\ a_{y\,\min} \le a_y \le a_{y\,\max} \\ a_{z\,\min} \le a_z \le a_{z\,\max} \end{aligned}\right\} \Rightarrow \mathbf{a}_{\min} \le \mathbf{a} \le \mathbf{a}_{\max} \tag{5.5}$$

A better way to express torque demand is through the control signal which is proportional to the actuation torque in torque or current controlled drives. A simple open loop model of the drive system can be constructed with the identified control signal equivalent inertia (*m*), viscous damping (*b*) and Coulomb friction (d_{coul}) parameters, from Section 4.5, to describe the control signal (u_t).

$$\left.\begin{aligned} u_{tx} &= m_x a_x + b_x v_x + d_{coulx}\,\mathrm{sgn}(v_x) \\ u_{ty} &= m_y a_y + b_y v_y + d_{couly}\,\mathrm{sgn}(v_y) \\ u_{tz} &= m_z a_z + b_z v_z + d_{coulz}\,\mathrm{sgn}(v_z) \end{aligned}\right\} \Rightarrow \mathbf{u}_t = \mathbf{Ma} + \mathbf{Bv} + \mathbf{D}_{coul}\,\mathrm{sgn}(\mathbf{v}) \tag{5.6}$$

Then, bounds are imposed on the control signal such that actuator saturation is avoided, as in the following equation:

$$\left.\begin{aligned} u_{tx\,\min} \le u_{tx} \le u_{tx\,\max} \\ u_{ty\,\min} \le u_{ty} \le u_{ty\,\max} \\ u_{tz\,\min} \le u_{tz} \le u_{tz\,\max} \end{aligned}\right\} \Rightarrow \mathbf{u}_{t\,\min} \le \mathbf{u}_t \le \mathbf{u}_{t\,\max} \tag{5.7}$$

Lastly, axis jerk must be limited for several reasons. Excessive jerk affects the tracking performance of the axis drives as high frequency motion commands result in poor tracking. Poor tracking can translate into inaccurate contouring, especially if the bandwidths of the drives are different and significantly lower than necessary. Furthermore, the jerk represents the frequency content of the commanded acceleration. High frequency content in the acceleration can excite vibrations in the machine tool structure which also degrades the total positioning. Finally, limits on the jerk help to ensure smooth motion. Hence, the jerk of each axis is bounded by a minimum and maximum value.

$$\left.\begin{array}{l} j_{x\,\min} \le j_x \le j_{x\,\max} \\ j_{y\,\min} \le j_y \le j_{y\,\max} \\ j_{z\,\min} \le j_z \le j_{z\,\max} \end{array}\right\} \Rightarrow \mathbf{j}_{\min} \le \mathbf{j} \le \mathbf{j}_{\max} \tag{5.8}$$

Overall the optimization problem that is solved in this thesis is expressed with the mathematical formulation in Equation (5.9). Here, t_k is the absolute time boundary between the *k*th and *k+1*th segments.

$$\begin{array}{l} \text{Objective Function: } \max F_k\,,\quad 1 \le k \le N \\ \text{Subject to:} \\ \begin{array}{ccccc} 0 & < & F_i & \le & F_{\max} \\ \mathbf{v}_{\min} & \le & \mathbf{v}(t) & \le & \mathbf{v}_{\max} \\ \mathbf{u}_{t\min} & \le & \mathbf{u}_t(t) & \le & \mathbf{u}_{t\max} \\ \mathbf{j}_{\min} & \le & \mathbf{j}(t) & \le & \mathbf{j}_{\max} \end{array} \quad \text{for} \quad t_{k-1} \le t \le t_k \end{array} \tag{5.9}$$

The solution methodologies of the worst-case optimization technique and the new heuristic optimization strategy are presented in the next section.

5.3 Solution Methodology

5.3.1 Worst-case Technique

In the worst-case optimization technique which was presented by Weck et al. [31], the maximum allowable feedrate for each segment is computed using the worst-case curvature of the given toolpath geometry, and worst case assumptions for unknown variables. Here, this method is briefly reviewed to allow comparison with the proposed heuristic technique. For a spline toolpath described with the parametric function, $\mathbf{C}(u) = [x \quad y \quad z]^T$, and a feed motion spline, $s(t)$, the axis velocities $\dot{\mathbf{C}}(t)$, accelerations $\ddot{\mathbf{C}}(t)$, and jerks $\dddot{\mathbf{C}}(t)$ can be expressed in terms of the geometric derivatives, and derivatives of the feed motion spline by applying the chain rule. The geometric derivatives, which are the derivatives with respect to the arc-length, are defined in Equation (5.10), and can be solved for in terms of the spline's parametric derivatives, and the derivatives of the feed correction polynomial, $u(s)$, which was presented in Section 3.3.

$$\frac{d\mathbf{c}}{ds} = \mathbf{C}_s = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}, \quad \frac{d^2\mathbf{C}}{ds^2} = \mathbf{C}_{ss} = \begin{bmatrix} x_{ss} \\ y_{ss} \\ z_{ss} \end{bmatrix}, \quad \frac{d^3\mathbf{C}}{ds^3} = \mathbf{C}_{sss} = \begin{bmatrix} x_{sss} \\ y_{sss} \\ z_{sss} \end{bmatrix}$$

$$\begin{aligned} \mathbf{C}_s &= \frac{d\mathbf{C}}{du}\frac{du}{ds} \\ \mathbf{C}_{ss} &= \frac{d\mathbf{C}}{du}\frac{d^2u}{ds^2} + \frac{d^2\mathbf{C}}{du^2}\left(\frac{du}{ds}\right)^2 \\ \mathbf{C}_{sss} &= \frac{d\mathbf{C}}{du}\frac{d^3u}{ds^3} + 3\frac{d^2\mathbf{C}}{du^2}\frac{d^2u}{ds^2}\frac{du}{ds} + \frac{d^3\mathbf{C}}{du^3}\left(\frac{du}{ds}\right)^3 \end{aligned} \tag{5.10}$$

Axis velocities $\mathbf{v} = \dot{\mathbf{C}}$, acceleration $\mathbf{a} = \ddot{\mathbf{C}}$, and jerks $\mathbf{j} = \dddot{\mathbf{C}}$ are thus:

$$\dot{\mathbf{C}} = \mathbf{C}_s \dot{s} \tag{5.11}$$

$$\ddot{\mathbf{C}} = \mathbf{C}_s \ddot{s} + \mathbf{C}_{ss} \dot{s}^2 \tag{5.12}$$

$$\dddot{\mathbf{C}} = \mathbf{C}_s \dddot{s} + 3\mathbf{C}_{ss} \ddot{s}\dot{s} + \mathbf{C}_{sss} \dot{s}^3 \tag{5.13}$$

Using Equation (5.11), the maximum feed due to the limits on the axis velocities can be derived. The maximum allowable feedrate due to velocity limits is found by substituting in the maximum velocity bounds, solving for the feed for each axis limit, and finally taking the minimum value of the feeds determined by all three axes, as in Equation (5.14). It is assumed that the bounds are symmetric, hence only the maximum velocity value and absolute values of the geometric tangent are required. Note that $[x_s \quad y_s \quad z_s]^T$ is the unit tangent vector and is evaluated at several points along the toolpath segment to obtain a feed limit profile. If any of the components of the unit tangent vector is equal to zero, then a feed limit is not imposed by the corresponding axis.

$$f_{vel} \le \min\left(\frac{v_{x\max}}{|x_s|}, \frac{v_{z\max}}{|y_s|}, \frac{v_{z\max}}{|z_s|} \right) \tag{5.14}$$

The feed limit due to the acceleration constraints is derived using Equation (5.12). Tangential acceleration is substituted with a worst-case (highest) value, $A_{\max}$, and axis acceleration is replaced with the specified axis acceleration limits. Hence solving for the feed, the feed limit due to the acceleration constraints is found as follows:

$$f_{acc} \le \min\left(\sqrt{\frac{a_{x\max} - |x_s| A_{\max}}{|x_{ss}|}}, \sqrt{\frac{a_{y\max} - |y_s| A_{\max}}{|y_{ss}|}}, \sqrt{\frac{a_{z\max} - |z_s| A_{\max}}{|z_{ss}|}} \right) \tag{5.15}$$

Note that $[x_{ss} \quad y_{ss} \quad z_{ss}]^T$ is the curvature vector and is also evaluated at several points along the toolpath. If any component of the curvature vector is zero, then the corresponding axis does not limit the feedrate due to acceleration constraints. Otherwise, in order to obtain a real positive solution for the feed, it is assumed that the axis acceleration limits are greater than the tangential acceleration limit, for example, $a_{x\max} > A_{\max}$, since the unit tangent vector components are less than or equal to one.

If a dynamic model of the drive system is available, the axis acceleration limits can be calculated based on axis torque limits. Considering the open loop model described in Equation (5.6), the maximum command acceleration can be obtained by replacing the actuation torque by the maximum torque limit and by assuming the maximum value for the axis velocity term. Solving for the axis acceleration yields:

$$\begin{aligned} a_{x\max} &= (u_{tx\max} - b_x v_{x\max} - d_{coulx}) / m_x \\ a_{y\max} &= (u_{ty\max} - b_y v_{y\max} - d_{couly}) / m_y \\ a_{z\max} &= (u_{tz\max} - b_z v_{z\max} - d_{coulz}) / m_z \end{aligned} \tag{5.16}$$

The axis acceleration limits based on the torque limits can then be substituted into Equation (5.15) to obtain the feed limit due to the torque constraints. Here, the control signal limits must yield acceleration limits that are greater than the tangential acceleration limit in order to obtain a real positive solution.

Finally, the feed limit due to the jerk constraints is derived using Equation (5.13). Jerk and acceleration in the feed direction are substituted with worst-case values, $A_{\max}$ and $J_{\max}$, and the axis jerk term is replaced with the maximum axis jerk bound. A cubic equation results for each axis, as shown in Equation (5.17). Solving for the roots will yield a feed limit for each axis. The lowest feed among the three axes is the overall feed limit due to the jerk constraints.

$$\begin{aligned} 0 &= |x_{sss}| f_{x,jerk}^3 + 3|x_{ss}| A_{\max} f_{x,jerk} + (|x_s| J_{\max} - j_{x\max}) \\ 0 &= |y_{sss}| f_{y,jerk}^3 + 3|y_{ss}| A_{\max} f_{y,jerk} + (|y_s| J_{\max} - j_{y\max}) \\ 0 &= |z_{sss}| f_{z,jerk}^3 + 3|z_{ss}| A_{\max} f_{z,jerk} + (|z_s| J_{\max} - j_{z\max}) \\ f_{jerk} &\le \min(f_{x,jerk}, f_{y,jerk}, f_{z,jerk}) \end{aligned} \tag{5.17}$$

Similarly, it is assumed that the axis jerk limits are greater than the tangential jerk limit, for example $j_{x\max} > J_{\max}$, in order to obtain a real positive feed. Non-real solutions are discarded and in the case that there exist three real solutions, the smallest positive value is taken as the feed limit.

The feed limit profile consists of the lowest feed limit among all of the velocity, acceleration or torque, and jerk constraints.

$$f_{\lim} \le \min(f_{vel}, f_{acc}, f_{jerk}) \tag{5.18}$$

Then considering each segment, the smallest feed limit to occur in the length of a segment is the final command feed, F_k. Using the optimized command feed values, the feed profile can be generated using the feed modulation strategy that was presented in Chapter 4. An example of the worst-case feed profile is illustrated in Figure 5-1. The worst-case feed optimization technique is simple and computationally inexpensive. However, the resulting feed profile is more conservative than necessary because of the assumption of worst-case values for unknown variables such as the tangential acceleration and jerk.

The proposed strategy in this thesis aims to create feed profiles with shorter cycle times compared to the worst-case technique, with an efficient search method that finds higher feedrates. Assumed worst-case values are replaced with the actual values of the feed motion profile. An example of an optimized feed profile obtained with the proposed heuristic strategy, which is explained in the following section, is also illustrated in Figure 5-1, and shows that the heuristic feed is generally higher than the worst-case feed.

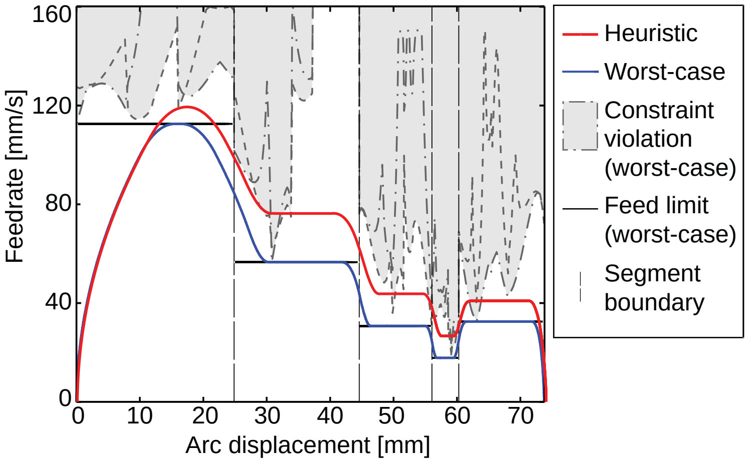

Figure 5-1. Comparison of worst-case feed optimization [31] to the proposed heuristic strategy.

5.3.2 Heuristic Strategy

The developed heuristic strategy, like the worst case technique, is general and can be applied to NURBS [26], [27] as well as other parametric toolpaths [2] [3] [9]. To solve the feed optimization problem presented in Section 5.2, the algorithm first narrows the search space to kinematically compatible feeds. Then, a rough scan of that range is performed to find a feasible solution. A feasible solution is defined as a command feedrate which results in a feed profile that satisfies all of the optimization constraints, listed in Equation (5.9), throughout the segment. Once a feasible feed is found, a bisection search method finds the highest feasible feed, which is defined as the optimized feed. In general, the algorithm consists of the following two parts, which are iterated one after another:

1. Selecting kinematically compatible feed candidates;

2. Checking for constraint violations along the trajectory.

Feed Selection

The algorithm assumes that the feed profile is generated using piecewise constant jerk values, leading to S-curve type feed transitions. Denoting the nominal feeds of the prior and current segments as F_{k-1} and F_k, the feed value at the segment boundary is $(F_{k-1}+F_k)/2$. The feed increment, Δ_k is defined as $\Delta_k = F_k - F_{k-1}$. The feed, tangential acceleration and jerk values are also bounded ($|\dot{s}| \le F_{\max}$, $|\ddot{s}| \le A_{\max}$ and $|\dddot{s}| \le J_{\max}$, where s is the arc displacement). Considering the *k*th segment with an arc-length of L_k, from Equation (4.23), the distance traveled at constant feed ($L_{f,k}$) in that segment is written again in Equation (5.19).

$$
\begin{aligned}
L_{f,1} &= L_1 - \left[\frac{\Delta_1^2}{2A_1} + \frac{A_1 \Delta_1}{2J_1} - \frac{3\Delta_2^2}{8A_2} + \frac{A_2^3}{24J_2^2} + \frac{F_1 A_2}{2J_2} + \frac{F_2 \Delta_2}{2A_2} \right] \geq 0 \\
L_{f,k} &= L_k - \begin{bmatrix} \dfrac{3\Delta_k^2}{8A_k} - \dfrac{A_k^3}{24J_k^2} + \dfrac{F_k A_k}{2J_k} + \dfrac{F_{k-1} \Delta_k}{2A_k} \\ -\dfrac{3\Delta_{k+1}^2}{8A_{k+1}} + \dfrac{A_{k+1}^3}{24J_{k+1}^2} + \dfrac{F_k A_{k+1}}{2J_{k+1}} + \dfrac{F_{k+1} \Delta_{k+1}}{2A_{k+1}} \end{bmatrix} \geq 0 \\
L_{f,N} &= L_N - \begin{bmatrix} \dfrac{3\Delta_N^2}{8A_N} - \dfrac{A_N^3}{24J_N^2} + \dfrac{F_N A_N}{2J_N} + \dfrac{F_{N-1} \Delta_N}{2A_N} \\ -\dfrac{\Delta_{N+1}^2}{2A_{N+1}} - \dfrac{A_{N+1} \Delta_{N+1}}{2J_{N+1}} \end{bmatrix} \geq 0
\end{aligned}
\qquad (5.19)
$$

The above kinematic compatibility conditions are used to determine the minimum and maximum reachable feeds between adjacent segments, and therefore, they dictate the heuristic search space. Given this range of kinematically compatible solutions, a set of rules are followed to obtain an optimized feedrate that satisfies the optimization constraints. In the following, the feed selection rules are explained with a 5-segment example, as illustrated in Figure 5-2.

At the start of the toolpath (Figure 5-2a), the search space is bounded with an initial feasible slow feed (e.g. 10 mm/s) and the maximum feed is found with a bisection search algorithm, which is illustrated in Figure 4-6, that utilizes Equation (5.19) to determine kinematic compatibility. After the search space has been defined by the upper and lower bounds $F_{k,\min}$ and $F_{k,\max}$, the algorithm iteratively tries out feed values to find the highest feedrate possible $F_{k,opt}$, which satisfies all of the optimization constraints. A bisection method - similar to the one used to find the kinematically compatible feeds - iteratively refines the search space and generates candidate feeds which bisect the refined search space, f_{mid}, as shown in Figure 5-3. Each candidate, which is inherently higher than the latest

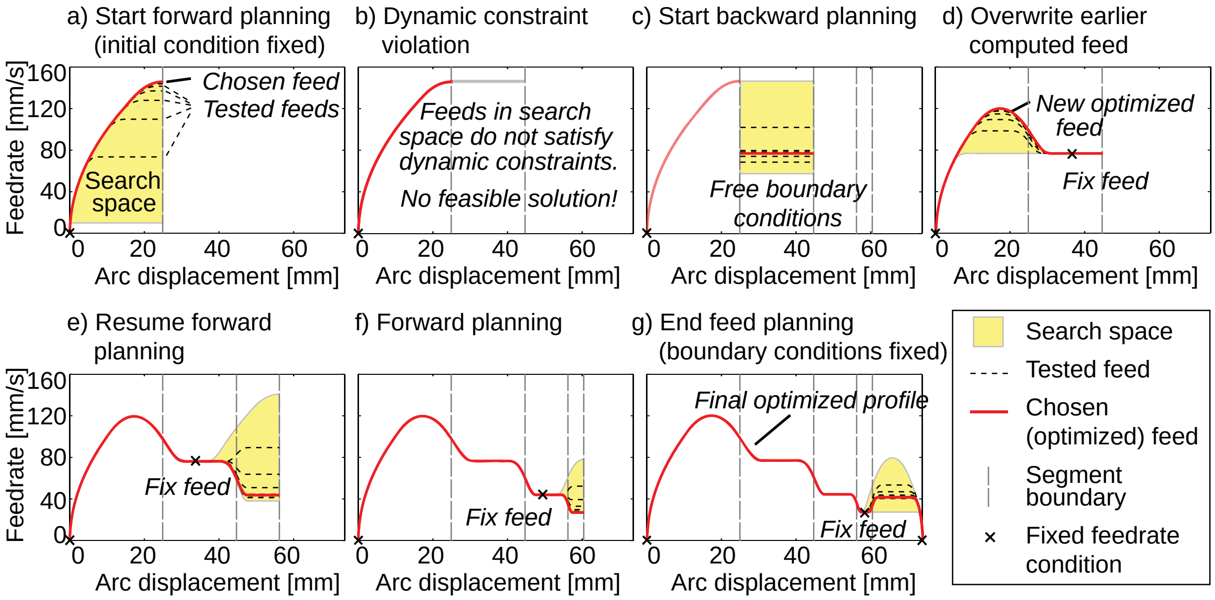

Figure 5-2. Proposed heuristic feed optimization technique (demonstrated with an example).

feasible solution, is tested against the optimization constraints, in Equation (5.9), to determine feasibility. If it does not violate them, it is stored as the latest feasible solution. The latest feasible solution at the end of the *n*th iteration is the optimized command feedrate.

Moving to the next segment, if a feasible solution that satisfies the optimization constraints in the search space cannot be found, as is the case in Figure 5-2b, then backward planning is performed to adjust the earlier feed values. The new search space, shown in Figure 5-2c, is bounded from below by the worst-case feed computed using Equations (5.14)-(5.18), considering the highest curvature in that segment, and maximum possible magnitudes for tangential acceleration and jerk ($|\ddot{s}| = A_{\max}$ and $|\dddot{s}| = J_{\max}$). The upper bound is set as the minimum feed reachable from the previous segment. The end conditions are freed such that the actual tangential acceleration and jerk profiles are zero (i.e. $\ddot{s} = 0$ and $\dddot{s} = 0$) and a constant feed profile is tested against the optimization constraints. A bisection search method is used to find the highest feasible feed candidate, and backward planning continues into earlier segment(s), as shown in Figure 5-2d, until a seamless connection can be made with the feed profile that was planned earlier in the forward pass.

It can be seen that the search space in the *1*st segment has been reduced, compared to the one in Figure 5-2a, and is bounded by the new fixed feedrate from below, and the maximum

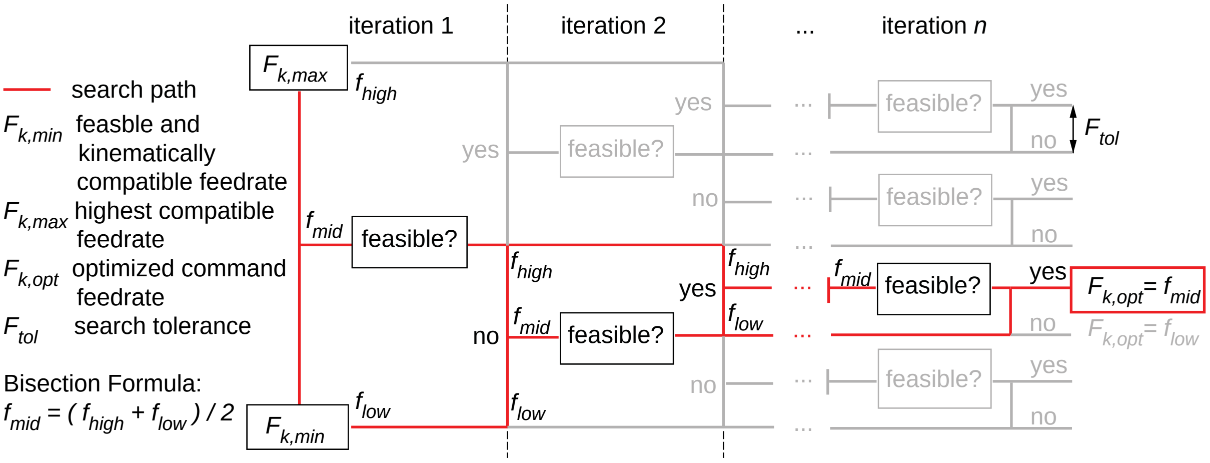
iteration 1
iteration 2
...
iteration *n*
search path
$F_{k,min}$ feasble and kinematically compatible feedrate
$F_{k,max}$ highest compatible feedrate
$F_{k,opt}$ optimized command feedrate
F_{tol} search tolerance
Bisection Formula:
$f_{mid} = (f_{high} + f_{low}) / 2$
$F_{k,max}$
$F_{k,min}$
f_{high}
f_{mid}
f_{low}
feasible?
yes
no
F_{tol}
$F_{k,opt} = f_{mid}$
$F_{k,opt} = f_{low}$

Figure 5-3. Bisection search algorithm to find an optimized feasible feed.

feed reachable from the *2*nd segment from above. Since the new fixed feedrate is lower than the previously found optimized feed for the *1*st segment, the feed, tangential acceleration and tangential jerk profiles will also be generally lower in absolute terms than the originally planned feed motion profiles from the forward planning stage. That is,

$$\text{if } \dot{s}_{new} \le \dot{s}_{old}\,, \qquad \text{then } \left|\ddot{s}_{new}\right| \le \left|\ddot{s}_{old}\right|, \qquad \left|\dddot{s}_{new}\right| \le \left|\dddot{s}_{old}\right| \tag{5.20}$$

In general, the lower feed is also a solution that satisfies all the optimization constraints of the previous segment. Considering Equations (5.11)-(5.13), the axis velocity, acceleration and jerk will also be lower if the component geometric derivatives are either both positive or both negative. For example, if the x-axis geometric derivatives and the feed motion derivatives are all positive, then the overall velocity, acceleration and jerk in the x-direction will also be less than the maximum x-axis limit.

$$\begin{aligned}
&\text{Given}: \; x_s, x_{ss}, \dot{s}, \ddot{s}, \dddot{s} > 0 \\
&v_{x\max} \ge x_s \dot{s}_{old} > x_s \dot{s}_{new} \\
&a_{x\max} \ge x_s \ddot{s}_{old} + x_{ss} \dot{s}_{old}^2 > x_s \ddot{s}_{new} + x_{ss} \dot{s}_{new}^2 \\
&j_{x\max} \ge x_s \dddot{s}_{old} + 3x_{ss} \ddot{s}_{old} \dot{s}_{old} + x_{sss} \dot{s}_{old}^3 > x_s \dddot{s}_{new} + 3x_{ss} \ddot{s}_{new} \dot{s}_{new} + x_{sss} \dot{s}_{new}^3
\end{aligned} \tag{5.21}$$

However, if the geometric and feed motion derivatives are not all in the same direction then it is not guaranteed that the optimization constraints will remain satisfied. In the rare circumstance that a lower feed does not satisfy the optimization constraints due to the non-convexity of the problem, the segment is flagged. As yet there is no heuristic rule to mediate this situation, so the fixed feed of the adjacent segment is set as the command feed despite the optimization constraint violations and backward planning continues. Backward planning ends when the forward feed profile and the backward feed profile are kinematically compatible. When backward planning is complete, forward planning resumes again from the foremost segment (Figure 5-2e).

The algorithm steps through each segment one by one, (Figure 5-2f-g), fixing the solution found in the previous segments and constructing the search space by solving for a kinematically compatible upper feed bound and a feasible lower bound that satisfies the kinematic compatibility conditions and optimization constraints. The use of the kinematic compatibility conditions effectively narrows down the set of possible solutions and with the set of heuristic rules an initial feasible feed is determined. For the *k*th segment, it is first determined whether a constant feed profile, where the feed of the previous segment is held the same through to the next segment, satisfies the optimization constraints. If it does not, then a scan of the kinematically compatible feed range at equal increments is performed to find a feasible feed. The scan can be performed in sequential or random order where the number of feeds to test depends on the resolution of the scan. For a kinematically compatible feed range defined by the bounds f_{low} and f_{high}, and a resolution (df), the number of feeds to test (N_{scan}), and the feeds (f_i) are determined with the following equations:

$$\begin{aligned} &N_{scan} = round[(f_{high} - f_{low})/df] \\ &f_i = f_{high} - i(f_{high} - f_{low})/N_{scan} \qquad i = 0,1,\ldots,N_{scan} \end{aligned} \tag{5.22}$$

Either the interval size or the number of intervals can be specified arbitrarily. In general, a feed resolution of 10 mm/s would be used. Then the number of feeds to test would be no more than 15 for a maximum range of zero to 150 mm/s. Once a feasible feed is found, it becomes the lower bound of the heuristic search space $(F_{k,\min})$ which is then used in the bisection search method in Figure 5-3, to find an optimized solution. If none of the test feeds prove to be feasible, then backward planning would be initiated.

In the last segment ($k = N$, $f_e = 0$), both initial and final conditions are fixed, as shown in Figure 5-3g, and the search space is bounded by the maximum reachable feed that can be decreased smoothly to zero by the end of the segment. That is, the feed compatibility condition for the final (*N*th) segment in Equation (5.19) is used, which already considers that the feed motion must come to a stop. Kinematically compatible feeds are those that leave enough travel length in the segment to decelerate to zero.

Constraint Evaluations

For the NURBS toolpaths developed in Chapter 3, constraint evaluations are performed using the NURBS formulation, kinematic equations, and the drives' dynamic model. If required, process-based bounds on the feedrate, which limit the cutting forces in the machining operation, can also be included. It follows from Equation (5.19) that the feed profile in the current segment is affected only by the preceding and next adjacent segments, thus making the feed planning in three consecutive segments independent from the rest of the toolpath. Hence, the constraints are checked only within a 3 segment window at a time (i.e. segments *k*-1, *k*, *k*+1). The NURBS toolpath position can be obtained as:

$$\mathbf{C}(u) = \begin{bmatrix} x(u) & y(u) & z(u) \end{bmatrix}^T = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \quad , \quad 0 \le u \le 1 \tag{5.23}$$

Recalling that, $\mathbf{P}_i$ are the control points, w_i are the weights, $n+1$ is the number of control points, p is the degree of the NURBS, and $N_{i,p}(u)$ denotes a basis function over the span i in the knot vector, $\mathbf{U} = \{u_0,\ldots,u_m\}$ [5] [26] [27]. A *7*th order feed correction polynomial (i.e. $u = f(s)$) is used for mapping the spline parameter (u) to the arc-length (s) with *2*nd order continuity at the connection boundaries, which mitigates unwanted feed fluctuations during interpolation and ensures acceleration continuous profiles to be generated, as explained in Chapter 3. Denoting the geometric derivatives with respect to the arc-length (Equation (5.10)) and applying the chain rule, the axis velocity, acceleration, and jerk profiles are derived as in Equations (5.11)-(5.13). The optimization constraints considered are the limits for axis velocity $\mathbf{v}$, control signal (i.e. actuation torque) $\mathbf{u}_t$, and jerk $\mathbf{j}$, which are expressed in Equation (5.9). Substituting in Equations (5.11)-(5.13) into the optimization constraints and normalizing them with respect to the limits yields the following 20 expressions in Equation (5.24).

$$
\begin{aligned}
-\dot{s} &< 0 \\
\dot{s} / F_{\max} - 1 &\leq 0 \\
1 - x_s \dot{s} / v_{x\min} &\leq 0 \\
1 - y_s \dot{s} / v_{y\min} &\leq 0 \\
1 - z_s \dot{s} / v_{z\min} &\leq 0 \\
x_s \dot{s} / v_{x\max} - 1 &\leq 0 \\
y_s \dot{s} / v_{y\max} - 1 &\leq 0 \\
z_s \dot{s} / v_{z\max} - 1 &\leq 0 \\
1 - (x_s \dddot{s} + 3x_{ss}\ddot{s}\dot{s} + x_{sss}\dot{s}^3) / j_{x\min} &\leq 0 \\
1 - (y_s \dddot{s} + 3y_{ss}\ddot{s}\dot{s} + y_{sss}\dot{s}^3) / j_{y\min} &\leq 0 \\
1 - (z_s \dddot{s} + 3z_{ss}\ddot{s}\dot{s} + z_{sss}\dot{s}^3) / j_{z\min} &\leq 0 \\
(x_s \dddot{s} + 3x_{ss}\ddot{s}\dot{s} + x_{sss}\dot{s}^3) / j_{x\max} - 1 &\leq 0 \\
(y_s \dddot{s} + 3y_{ss}\ddot{s}\dot{s} + y_{sss}\dot{s}^3) / j_{y\max} - 1 &\leq 0 \\
(z_s \dddot{s} + 3z_{ss}\ddot{s}\dot{s} + z_{sss}\dot{s}^3) / j_{z\max} - 1 &\leq 0 \\
1 - [m_x(x_s\ddot{s} + x_{ss}\dot{s}^2) + b_x z_s \dot{s} + d_{coulx}\,\mathrm{sgn}(x_s)] / u_{tx\min} &\leq 0 \\
1 - [m_y(y_s\ddot{s} + y_{ss}\dot{s}^2) + b_y y_s \dot{s} + d_{couly}\,\mathrm{sgn}(y_s)] / u_{ty\min} &\leq 0 \\
1 - [m_z(z_s\ddot{s} + z_{ss}\dot{s}^2) + b_z z_s \dot{s} + d_{coulz}\,\mathrm{sgn}(z_s)] / u_{tz\min} &\leq 0 \\
[m_x(x_s\ddot{s} + x_{ss}\dot{s}^2) + b_x x_s \dot{s} + d_{coulx}\,\mathrm{sgn}(x_s)] / u_{tx\max} - 1 &\leq 0 \\
[m_y(y_s\ddot{s} + y_{ss}\dot{s}^2) + b_y y_s \dot{s} + d_{couly}\,\mathrm{sgn}(y_s)] / u_{ty\max} - 1 &\leq 0 \\
[m_z(z_s\ddot{s} + z_{ss}\dot{s}^2) + b_z z_s \dot{s} + d_{coulz}\,\mathrm{sgn}(z_s)] / u_{tz\max} - 1 &\leq 0
\end{aligned}
\tag{5.24}
$$

Thus, in order to evaluate the constraints, the geometric derivatives at pre-selected points along the toolpath are obtained, and the feed, tangential acceleration and jerk values in the feed motion trajectory that occur at those points are also computed as required. Constraint evaluation check points are selected according to the variations in the toolpath segment as shown in Figure 5-4. Portions of the segments with large variations (i.e. high curvature) should contain more evaluation points than the portions with small variations (i.e. low curvature), to ensure acceptable enforcement of the constraints throughout the whole toolpath.

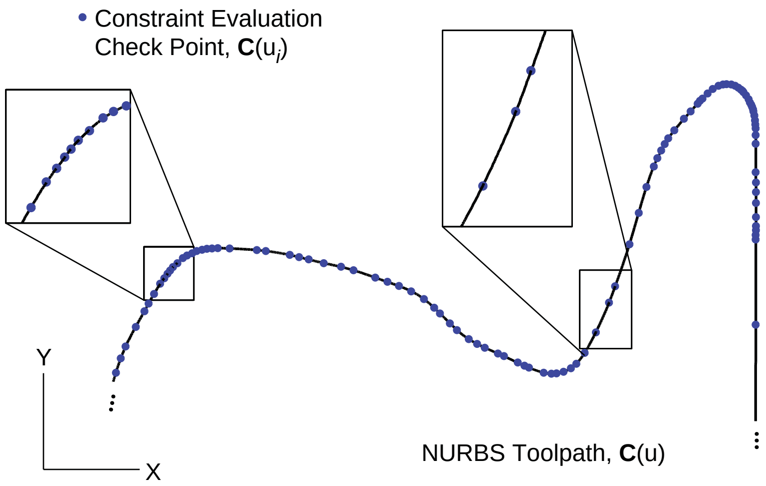

Figure 5-4. Constraint evaluation points based on the variation of the toolpath geometry.

The constraint evaluation points can be a subset of the points resulting from the adaptive integration of the segment length, determined during the application of Simpson's Rule in Section 3.3.1, where a large subset corresponds to tighter constraints and a small subset translates into relaxed constraints. They can also be selected based on the knot distribution in the NURBS segment, as the knots and their positions in the parametric space relate to the number and influence of the control points that shape the curve. In general the more control points there are, the more curvature variation occurs in the curve, thus resulting in more knots to act as constraint evaluation points. Geometric derivatives, $\mathbf{C}_s$, $\mathbf{C}_{ss}$, and $\mathbf{C}_{sss}$, of each constraint evaluation point are calculated prior to the feed optimization. In this thesis, a subset of the integration data points is used.

Feed $\dot{s}$, acceleration $\ddot{s}$, and jerk $\dddot{s}$ values at each constraint evaluation point are obtained during the feed optimization by solving for the relative time value that corresponds to the arc displacement of each point. Given a proposed feed profile, from the feed selection step, and the arc displacement (s) of a constraint evaluation point in a given segment, using Equations (4.11)-(4.13), the phase in which the arc displacement occurs is determined by comparing the given s value to the arc displacement reached at the end of each phase. For example, if $s_{4e} < s \leq s_{5e}$, then the constraint evaluation point lies within phase ⑤ of the feed motion profile of a segment. Then, the corresponding displacement equation is used to solve for the relative time value, τ, as shown in Equations (5.25)-(5.27) for the initial (*1*st), middle (*k*th) and final (*N*th) segments. The displacement equations for constant feed phases are linear and can therefore be rearranged to solve for the relative time variable. Quadratic functions, which describe the trajectory during constant acceleration phases are solved with the numerically stable quadratic formula, while cubic equations are solved using the Newton Raphson method, where the time duration of the constant jerk phase serves as the initial guess, τ_0, for the relative time parameter. Finally, using Equations (4.2)-(4.10), the feed, tangential acceleration and jerk values are computed at the calculated relative time values. Then the axis velocity, acceleration and jerk values that occur at the given constraint evaluation points are then calculated with Equations (5.11)-(5.13).

If any of the constraints are violated in a segment, the proposed feedrate is determined to be infeasible and becomes the upper bound of the refined search space for the next iteration. On the other hand, if all of the constraints are satisfied, then the proposed feed is stored as the latest feasible feed and becomes the lower bound of the refined search space.

$$\text{(1st)} \begin{cases} 0 \le s \le s_{1e} \Rightarrow 0 = \frac{1}{6}J_1\tau^3 + f_s\tau - s, \quad \tau_0 = T_{j,1} \\ s_{1e} < s \le s_{2e} \Rightarrow 0 = \frac{1}{2}A_1\tau^2 + f_{1e}\tau + s_{1e} - s \\ s_{2e} < s \le s_{3e} \Rightarrow 0 = -\frac{1}{6}J_1\tau^3 + \frac{1}{2}A_1\tau^2 + f_{2e}\tau + s_{2e} - s, \quad \tau_0 = T_{j,1} \\ s_{3e} < s \le s_{4e} \Rightarrow 0 = f_{3e}\tau + s_{3e} - s \\ s_{4e} < s \le s_{5e} \Rightarrow 0 = \frac{1}{6}J_2\tau^3 + f_{4e}\tau + s_{4e} - s \\ s_{5e} < s \le s_{6e} \Rightarrow 0 = \frac{1}{2}A_2\tau^2 + f_{5e}\tau + s_{5e} - s, \quad \tau_0 = T_{j,2} \end{cases} \tag{5.25}$$

$$\text{(}k\text{th)} \begin{cases} 0 < s \le s_{1e} \Rightarrow 0 = \frac{1}{2}A_k\tau^2 + f_s\tau - s \\ s_{1e} < s \le s_{2e} \Rightarrow 0 = -\frac{1}{6}J_k\tau^3 + \frac{1}{2}A_k\tau^2 + f_{1e}\tau + s_{1e} - s, \quad \tau_0 = T_{j,k} \\ s_{2e} < s \le s_{3e} \Rightarrow 0 = f_{2e}\tau + s_{2e} - s \\ s_{3e} < s \le s_{4e} \Rightarrow 0 = \frac{1}{6}J_{k+1}\tau^3 + f_{3e}\tau + s_{3e} - s, \quad \tau_0 = T_{j,k+1} \\ s_{4e} < s \le s_{5e} \Rightarrow 0 = \frac{1}{2}A_{k+1}\tau^2 + f_{4e}\tau + s_{4e} - s \end{cases} \tag{5.26}$$

$$(N\text{th})\begin{cases} 0 < s \le s_{1e} \Rightarrow 0 = \frac{1}{2}A_N\tau^2 + f_s\tau - s \\ s_{1e} < s \le s_{2e} \Rightarrow 0 = -\frac{1}{6}J_N\tau^3 + \frac{1}{2}A_N\tau^2 + f_{1e}\tau + s_{1e} - s, \quad \tau_0 = T_{j,N} \\ s_{2e} < s \le s_{3e} \Rightarrow 0 = f_{2e}\tau + s_{2e} - s \\ s_{3e} < s \le s_{4e} \Rightarrow 0 = \frac{1}{6}J_{N+1}\tau^3 + f_{3e}\tau + s_{3e} - s, \quad \tau_0 = T_{j,N+1} \\ s_{4e} < s \le s_{5e} \Rightarrow 0 = \frac{1}{2}A_{N+1}\tau^2 + f_{4e}\tau + s_{4e} - s \\ s_{5e} < s \le s_{6e} \Rightarrow 0 = -\frac{1}{6}J_{N+1}\tau^3 + \frac{1}{2}A_{N+1}\tau^2 + f_{5e}\tau + s_{5e} - s, \quad \tau_0 = T_{j,N+1} \end{cases} \tag{5.27}$$

Iterations

When processing each segment, the candidate feed generation and constraint evaluation steps are iterated one after another, which helps to refine the results. During each iteration, a new candidate feed value is generated by bisecting the lower and higher bound of the refined search space. If the candidate feed does not violate the optimization constraints in Equation (5.24), then it becomes the new lower bound of the search space. Conversely, if it violates any constraint, then the candidate feed replaces the higher bound. This allows the algorithm to zone in on a possibly better solution, if there is one, without wasting valuable computational time. However, this is subject to the underlying assumption that the solution space is convex. For example, if there are two feasible feeds, f_1 and f_2 such that $f_1 < f_2$, then there exists a third feed in between the two feeds, such that $f_1 < f_3 < f_2$, that is also feasible. The iteration stops when the refined feasible feed is within a given tolerance, F_{tol}, of the next feed candidate, as shown in Figure 5-3. In the implementation, it was found that using a search tolerance of 1 mm/s, for the experimental setup where the maximum axis velocity is 150 mm/s, led to successful cycle time reduction, without becoming overburdening in terms of computational load. The resulting number of bisection operations in solving for the best solution with a range defined by $F_{k,\min}$ and $F_{k,\max}$ can be calculated with the following equation:

$$N_{iter} = round[\log_2((F_{k,\max} - F_{k,\min}) / F_{tol})] + 1 \tag{5.28}$$

Sample calculations using Equation (5.28) are listed in Table 5-1.

Table 5-1. Sample calculation of the number of bisection operations for a range defined by $F_{k,\min}$=0 and $F_{k,\max}$=100 mm/s.

Search Tolerance, F_{tol} [mm/s]	Number of Bisection Operations, N_{iter}
2	7
1	8
0.1	11
0.01	14

One of the principal advantages of the proposed technique is that the duration required to process each segment can be deterministically specified, by setting the feed scanning resolution (df), search tolerance (F_{tol}), and the number of constraint evaluations to be performed for each segment. This makes the algorithm highly suitable for real-time implementation in comparison to gradient search-based methods [32]. The heuristic rules ensure that a feasible feed is computed, within the allowed computational window. A look-ahead buffer is still required in order to plan sufficient distance for accelerations and decelerations, which also holds for other feed optimization techniques as well [30], [31], [32]. Although the computational duration becomes nondeterministic when backward planning is invoked, an upper bound on the number of blocks that may need to be processed in one cycle can be computed by considering the smallest possible segment size and the kinematic properties of the feed profile as shown in Section 4.4. By setting the look-ahead buffer to be sufficiently large and utilizing a fast enough CPU, the problem of NC instruction overrun can be avoided, as is done in current CNC interpolators.

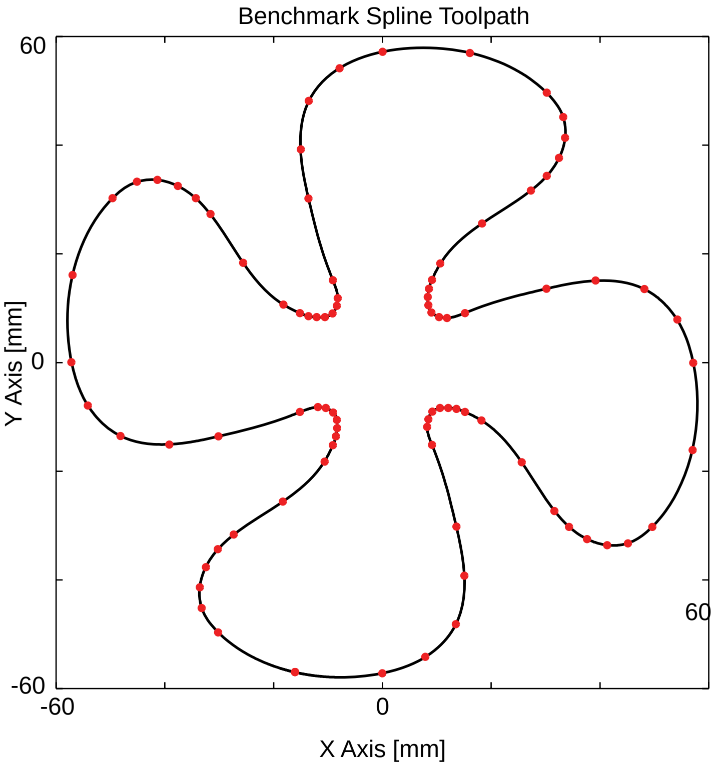

Figure 5-5. Benchmark contour - 88-segment quintic spline toolpath [9].

5.4 Experimental Results

The heuristic feed optimization strategy has been validated experimentally on the 3-axis router that was introduced in Section 4.5. Adaptive sliding mode control is used for closing the servo loop. The loop closure (interpolation) period was 1 ms. The maximum velocity, jerk, and control signal (i.e. actuation torque) limits that were used are 150 mm/s, 25,000 mm/s^3, and 5 V (50% actuation torque), respectively. The first benchmark is realized by comparing the heuristic strategy with the worst-case [31] and gradient-based [32] solutions, using a fan-shaped quintic spline toolpath from [9] with 150% scaling **(**Figure 5-5**).**

The kinematic profiles obtained with all three methods are summarized in Figure 5-6**.**

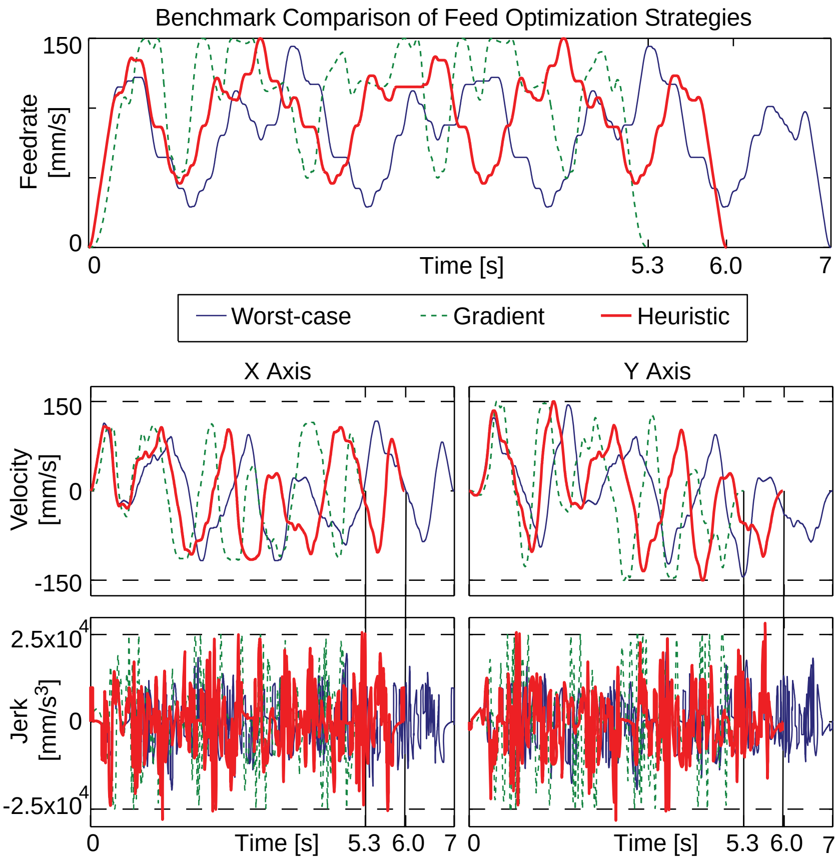

Figure 5-6. Optimized feedrate profiles using worst-case [31], gradient-based [32], and heuristic (proposed) optimization strategies

While implementing the worst-case method, jerk constraints were incorporated using Equation (5.17) and assuming maximum possible magnitudes for tangential acceleration and jerk. All three methods were applied subject to the same constraints on the feedrate, axis velocity, torque demand, and jerk, as defined in the feed optimization problem formulation in Equation (5.9). Each method satisfies the feedrate and axis velocity constraints. However, in the case of the axis jerk constraint there are instances where the proposed heuristic method violates the prescribed limit by small amounts. This was observed to occur because the number of constraint evaluation points used in the heuristic method was less than the number of evaluation points in the gradient method. This resulted in a slight relaxation of the limits, allowing for minor violations to occur. The violations were found to be tolerable as they did not adversely affect the tracking performance of the controlled system, as seen in Figure 5-7. Additionally, the real-time control signal, which directly represents the actuation torque, also slightly exceeds the prescribed limits because of the dynamic nature of the closed-loop

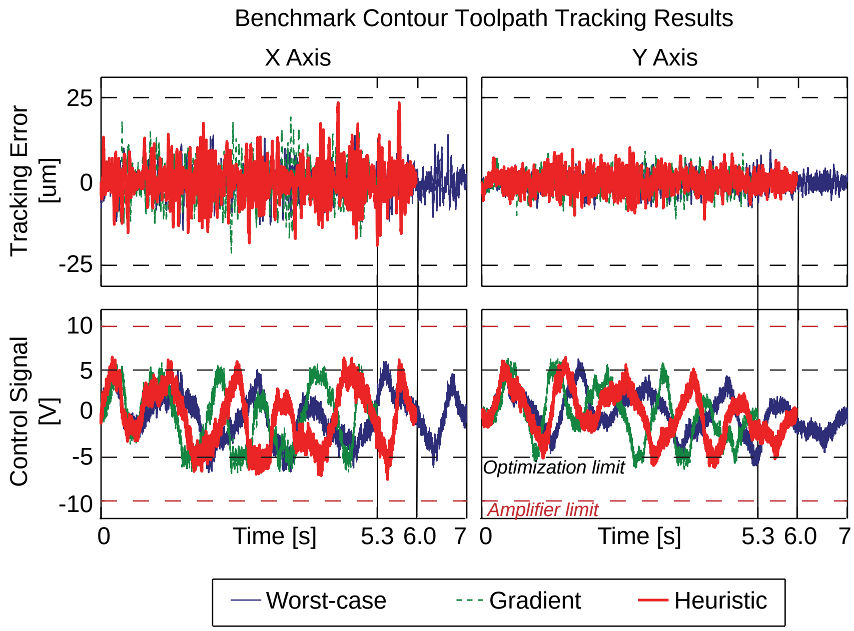

Figure 5-7. Comparison of tracking performance for different optimization strategies.

feedback servo controller which was not considered in the dynamic model for torque demand. The problem formulation for all three methods used an open loop dynamic model to predict the torque demand. For this reason the control signal limit was set rather conservatively (50% of actuation torque) so that the amplifier limit which occurs at 10 [V] is never invoked as seen in Figure 5-7. Alternatively, the limits could be modified according to the variance of the closed-loop control signal, which was reported in [41].

Table 5-2. Computational time for benchmark toolpath feed optimization.

Optimization Strategy	Computation Time [s]
Worst-case	1.555
Heuristic	6.761
Gradient-based	69.415

The proposed heuristic method exhibits a 13.73% decrease in cycle time compared to the worst-case method from 6.975 s to 6.017 s and maintains a comparable tracking accuracy. The computational time used by each optimization technique is summarized in Table 5-2. The implementation was made on a Pentium IV 3 GHz computer using MATLAB. The computational time is an order of magnitude less, compared to the gradient-based feed

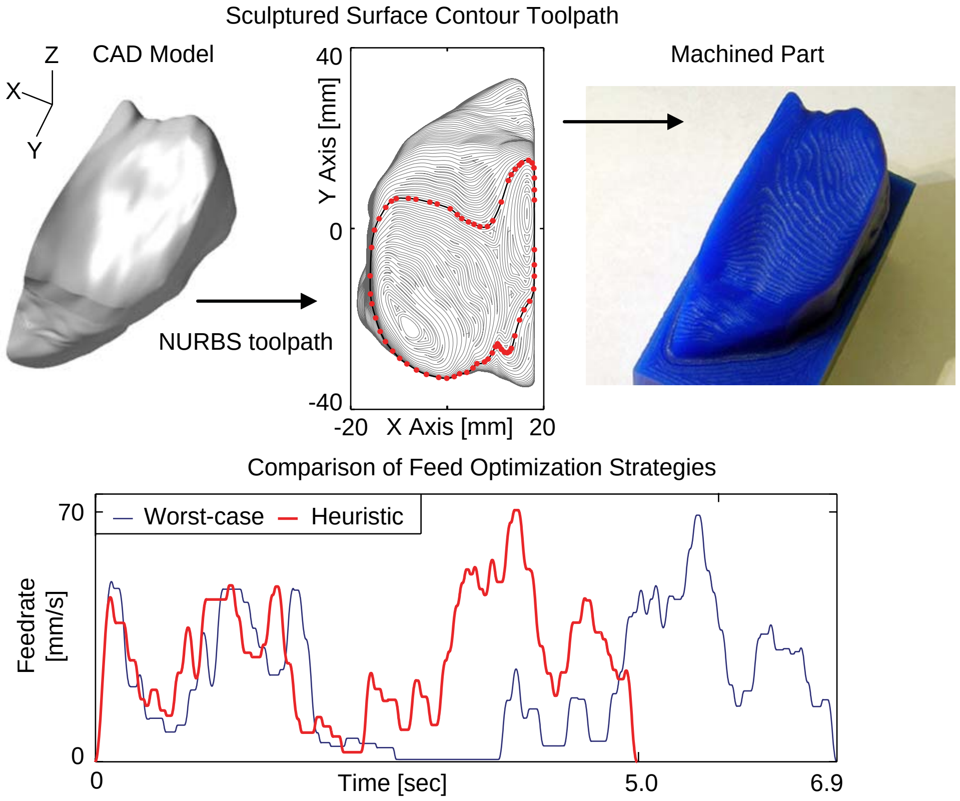

Figure 5-8. Optimized feedrate profiles of a sample NURBS toolpath.

optimization, and about four times longer compared to the worst-case feed optimization. The heuristic feed optimization strategy provides a good compromise between the worst-case and gradient based methods. On the one hand, the computational load is comparable to the worst-case technique and the cycle time is shorter. On the other hand, the cycle time is longer than the gradient-based solution, because there is less restriction on the shape of the feed profile in the latter approach. The gradient-based technique doesn't require zeroed acceleration and jerk boundary conditions between segments, as the jerk-limited S-curve profile does. However, the simplified feed motion profile contributes to the reduction in the computational load, which is observed to be approximately a factor of ten.

The heuristic and worst-case techniques were also compared in surface machining the tibial-plateau (lower knee joint) bone implant, which was scaled by 200% for machining convenience. The CAD model was converted into contour NURBS after applying ball end tool offset compensation (Figure 5-8) as explained in Chapter 4, and subsequently machined out of wax using both optimization techniques. In order to accelerate the experiment, a

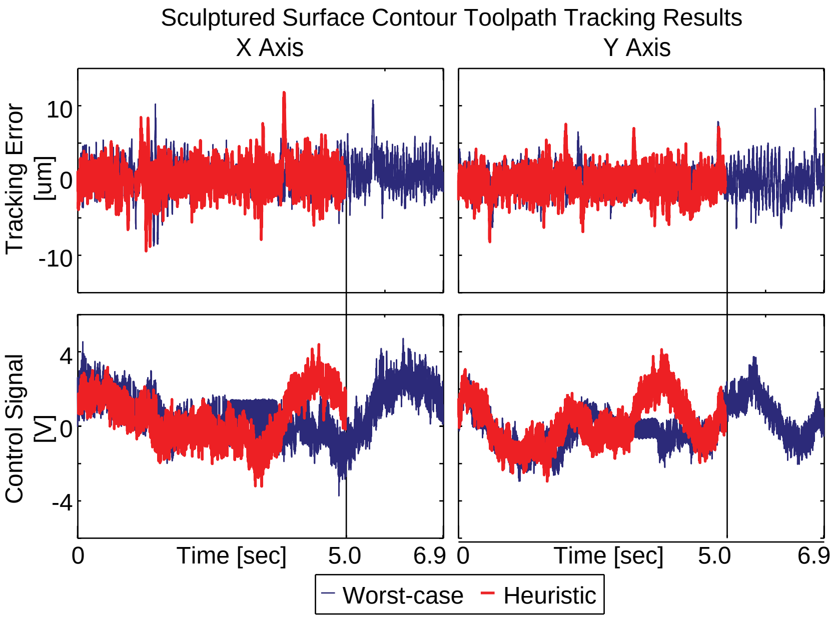

Figure 5-9. Tracking performance of contour machining for a biomedical implant.

relatively coarse toolpath tolerance (25 um) was used, which can be made tighter if a smoother surface is required. Additionally, the step down, that is the height between contour levels, was set to 0.5 mm.

Experimental results comparing the proposed heuristic strategy with the worst-case technique for the highlighted contour are shown in Figure 5-9 and a cycle time comparison for the representative contour and the complete toolpath is given in Table 5-3. Note that the duration of the total NURBS cycle time excludes linear tool movements. As seen, 26.36% time reduction is observed for the highlighted contour and an overall 21.62% cycle time reduction is realized for all of the NURBS toolpaths combined, compared to worst-case feed planning. The tracking error in both cases is less than 15 um, indicating that the cycle time reduction does not come at the expense of the drives' dynamic accuracy.

Table 5-3. Cycle time comparison for implant surface.

Optimization Strategy	Sample NURBS Toolpath [s]	All NURBS Toolpaths [s]
Worst-case	6.891	601.009
Heuristic	5.074	469.341

Hence, the heuristic feed optimization strategy yields shorter cycle times compared to the worst-case technique, while also preserving the desired machining tolerance.

5.5 Conclusions

This chapter has presented a new feedrate optimization strategy that can be applied to trajectory generation using spline toolpaths. The proposed strategy intelligently uses heuristic rules, alongside analytically derived feasibility conditions, to achieve lower cycle times compared to the widely accepted worst-case approach in NURBS toolpath feed planning. Cycle time reductions are shown to be as high as 26% from the worst-case solution. The computational load is approximately ten times less compared to gradient-based methods while being just four times longer than the worst-case implementation. The heuristic approach offers a compromise between the simplistic worst-case approaches, which are generally conservative, and the complex gradient-based techniques, which can be computationally expensive. Practically, the cycle time reductions are obtained at low computational cost, and the utilization of the S-curve as the basis feed function allows the heuristic technique to be implemented inside, or in conjunction with, existing CNC interpolators. Contour and surface machining experiments were performed to validate the proposed strategy. Experimental results show that the tracking error is maintained at less than 25 um, which indicates that the achieved cycle time reductions do not adversely affect the tracking performance of the drive system. The heuristic feed optimization strategy effectively minimizes the cycle time while maintaining the desired contouring accuracy of the machined part.

Chapter 6

Conclusions and Future Work

Overall, this thesis has presented a NURBS toolpath interpolation scheme with continuous feedrate modulation and feedrate optimization for CNC machining. The proposed techniques result in coordinated axis motion that is smooth and time-optimal within the constraints of the drives' dynamic limits.

NURBS toolpaths were parameterized with geometric curvature continuity using beta-constraints, which allowed for additional flexibility that parametric continuity constraints do not afford in shaping the curve to avoid oscillations. The beta shape parameters are a useful design tool, but as of yet there is no automatic way of assigning their values such that the parameterization always generates the desired smooth toolpath. It was found that smooth toolpaths could always be found, but sometimes required designer intervention in the toolpath planning, which is unfortunately time consuming. Several methods to improve the automation of the NURBS toolpath parameterization may include the addition of a jerk penalizing term into the curve fitting objective function, better heuristics for setting the values of beta parameters, and segmenting the data points adaptively according to favorable conditions for generating smooth non-oscillatory splines. Furthermore, the weights in the NURBS equation were underutilized in the NURBS curve fitting method, in this work as well as in the literature. Further research into the effect of weights and how they can be assigned values other than one to parameterize a better behaving spline is recommended.

For NURBS interpolation, the arc-lengths of the NURBS segments were integrated numerically with Simpson's adaptive quadrature method. When the quadrature data of each segment is summed cumulatively, it provided a discrete mapping of the spline parameter to the arc displacement along the spline segment. An inverse relationship which expresses the spline parameter in terms of arc displacement was obtained by solving a constrained least squares optimization problem with the Lagrange Multipliers technique. This function is known as the feed correction polynomial. Interpolation using the feed correction polynomial was shown to reduce unwanted feedrate fluctuations while being numerically efficient and robust against numerical errors. Feedrate fluctuations are reduced from around 40 % for natural interpolation to 0.1 % for interpolation with feed correction. Excessive acceleration and jerk in the axes are also avoided. However, the main challenge encountered in this work was ensuring that the feed correction polynomial accurately reflected the spline parameter and arc displacement relationship. This issue was addressed by evaluating the mean squared error (i.e. variance) between the actual data and its estimates and if the error was above a specified tolerance, dividing the data into two sets and using multiple splines. Most numerical instability which caused inaccurate fitting happened when a large change in the spline parameter resulted in only a very small change in arc displacement. Although using multiple splines was presented in this work, further investigation into possible numerical instabilities encountered is suggested for future work.

A generalized framework for feed modulation over multiple toolpath segments using the S-curve function was developed in this thesis. The feed modulation technique used analytically derived kinematic compatibility conditions to ensure that the displacement, feedrate and acceleration profiles were continuous and jerk-limited in all axes. The feed is modulated using a bisection search method that is simple and numerically efficient. With the use of a look-ahead window, the feed modulation method is well suited to support real-time interpolation. Moreover, the framework can be interfaced with different feed optimization techniques such as the worst-case curvature method [31]. It served as the foundation for the new heuristic feed optimization strategy developed in this thesis. Using analytically derived kinematic compatibility conditions and an efficient bisection search algorithm with optimization constraints to test the feasibility, the segment feed is maximized. It was shown that the new strategy reduced the cycle time by approximately 13% - 26% compared to the worst-case curvature approach and is only slightly more computationally intensive. The heuristic strategy has less computational load compared to a gradient-based solution [32] making it very practical to implement on a CNC controller.

The overall NURBS trajectory generator has been validated in machining experiments conducted on a 3-axis router. By ensuring that the toolpath trajectory has continuous acceleration and is jerk limited in all axes, it is demonstrated that the coordinated motion is smooth and continuous. Cycle time reductions are obtained and axis tracking errors do not exceed 25 um. Contouring accuracy is not sacrificed for faster feedrates. Therefore, the

NURBS trajectory generator can be practically and economically integrated into, or in conjunction with, existing CNC controllers to meet higher demands for high precision in faster cycle times.

Suggestions for future work include implementing the algorithms developed in this thesis in a real-time environment on the dSPACE controller, testing the algorithms with NURBS toolpaths that are parameterized by standard CAD/CAM packages or derived directly from the CAD geometry rather than from linear toolpaths, and investigating numerical instabilities in the feed correction polynomial curve fitting.

References

[1] Shpitalni, M., Koren, Y., Lo, C. C., 1994, Real-time curve interpolators, Computer-Aided Design, 26/11:832-838.

[2] Koren, Y., Lin, R.-S., 1995, Five-Axis Surface Interpolators, Annals of CIRP, 44/1:379-382.

[3] Lartigue, C., Tournier, C., Ritou, M., Dumur, D., 2004, High-Performance NC for HSM by means of Polynomial Trajectories, Annals of CIRP, 53/1:317-320.

[4] Langeron, J.M., Duc, E., Lartigue, C., Bourdet, P., 2004, A new format for 5-axis tool path computation, using Bspline curves, Computer-Aided Design, 36/12:1219-1229.

[5] Piegl, L., Tiller, W., 2003, *The NURBS book, 2nd Edition*; Springer-Verlag, Berlin.

[6] Barsky, B.A., DeRose, T.D., 1989, Geometric continuity of parametric curves: three equivalent characterizations, IEEE Computer Graphics and Applications 9/6:60-68.

[7] Barsky, B.A., DeRose, T. D., 1990, Geometric continuity of parametric curves: constructions of geometrically continuous splines, IEEE Computer Graphics and Applications, 10/1:60-68.

[8] Wang, F.-C., Yang, D.C.H., 1993, Nearly Arc-Length Parametrized Quintic-Spline Interpolation for Precision Machining, Computer-Aided Design, 25/5: 281-288.

[9] Wang, F.-C., Wright, P.K., Barsky, B.A., Yang, D.C.H., 1999, Approximately Arc-Length Parametrized C^3 Quintic Interpolatory Splines, ASME Journal of Mechanical Design, 121/3:430-439.

[10] Erkorkmaz, K., Altintas, Y., 2005, Quintic Spline Interpolation with Minimal Feed Fluctuation, ASME Journal of Manufacturing Science and Engineering, 127/2:339-349.

[11] Lee, R.S., Liang, S.P., 2006, A strain energy minimization method for generating continuous NURBS-based motion curves in free-form surface machining, International Journal of Advanced Manufacturing Technology, 28/11-12:1136-1145.

[12] Sencer, B., 2005, M.A.Sc. Thesis: Five-Axis Trajectory Generation Methods. The University of British Columbia, Vancouver.

[13] Huang, J.-T., Yang, D.C.H., 1992, Precision Command Generation for Computer Controlled Machines, Precision Machining: Technology and Machine Development and Improvement, ASME-PED, 58:89-104.

[14] Otsuki, T., Kozai, H., Wakinotani, Y., Fanuc Ltd., Yamanashi, Japan, 1998, Free-Form Curve Interpolation Method and Apparatus, United States Patent 5,815,401.

[15] Lin, R.S., 2000, Real-time surface interpolator for 3-D parametric surface machining on 3-axis machine tools, International Journal of Machine Tools and Manufacture, 40/10:1513-1526.

[16] Erkorkmaz, K., Altintas, Y., 2001, High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation, International Journal of Machine Tools and Manufacture, 41/9:1323-1345.

[17] Cheng, C.W., Tsai, M.C., Maciejowski, J., 2006, Feedrate control for Non Uniform rational B-spline motion command generation, in Proc. Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 220/B11: 1855-1861.

[18] Lei, W.T., Sung, M.P., Lin, L.Y., Huang, J.J., 2007, Fast real-time NURBS path interpolation for CNC machine tools, International Journal of Machine Tools and Manufacture, 47/10:1530-1541.

[19] Macfarlane, S., Croft, E.A., 2001, Design of Jerk Bounded Trajectories for On-Line Industrial Robot Applications, Proceedings IEEE International Conference on Robotics and Automation, 1:979-984.

[20] Erkorkmaz, K., 2004, Ph.D. Thesis: Optimal Trajectory Generation and Precision Tracking Control for Multi-Axis Machines. The University of British Columbia, Vancouver.

[21] Pritschow, G., 1997, Course notes: Steuerungtechnik der Werkzeugmaschinen und Industrieroboter (control techniques of machine tools and industrial robots), Institute of Control Technology for Machine Tools and Manufacturing Units, Stuttgart University, Germany.

[22] Lin, M.-T., Tsai, M.-S., Yau, H.-T., 2007, Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm, International Journal of Machine Tool and Manufacture, 47/15:2246-2262.

[23] Liu, X., Ahmad, F., Yamazaki, K., Mori, M., 2005, Adaptive interpolation scheme for NURBS curves with the integration of machining dynamics, International Journal of Machine Tools and Manufacture, 45/4-5:433-444.

[24] Xu, R.Z., Xie, L., Li, C.X., Du., D.S., 2008, Adaptive parametric interpolation scheme with limited acceleration and jerk values for NC machining, International Journal of Advanced Manufacturing Technology, 36/3-4:343-354.

[25] Park, J., Nam, S.H., Yang, M.Y., 2005, Development of a real-time trajectory generator for NURBS interpolation based on the two-stage interpolation method, International Journal of Advanced Manufacturing Technology, 26/4:359-365.

[26] Koninckx, B., Van Brussel, H., 2002, Real-Time NURBS Interpolator for Distributed Motion Control, Annals of the CIRP, 51/1:315-318.

[27] Brecher, C., Lange, S., Merz, M., Niehaus, F., Wenzel, C., Winterschladen, M., Weck, M., 2006, NURBS Based Ultra-Precision Free-Form Machining, Annals of the CIRP, 55/1:547-550.

[28] Bobrow, J.E., Dubowsky, S., Gibson, J.S., 1985, Time-Optimal Control of Robotic Manipulators Along Specified Paths, International Journal of Robotics Research, 4/3:3-17.

[29] Constantinescu, D., Croft, E.A., 2000, Smooth and Time-Optimal Trajectory Planning for Industrial Manipulators Along Specified Paths, Journal of Robotic Systems, 17/5:233-249.

[30] Dong, J., Ferreira, P.M., Stori, J.A., 2007, Feed-rate optimization with jerk constraints for generating minimum time trajectories, International Journal of Machine Tools and Manufacture, 47:1941-1955.

[31] Weck, M., Meylahn, A., Hardebusch, C., 1999, Innovative Algorithms for Spline-Based CNC Controller, Annals of the German Academic Society for Production Engineering, VI/1:83-86.

[32] Altintas, Y., Erkorkmaz, K., 2003, Feedrate Optimization for Spline Interpolation in High Speed Machine Tools, Annals of the CIRP, 52/1:297-302.

[33] Cheng, Y., Chin, J., 2003, Machining contour errors as ensembles of cutting, feeding and machine structure effects, International Journal of Machine Tools and Manufacture, 43/10:1001-1014.

[34] Kyriakopoulos, K.J., Saridis, G.N., 1994, Minimum Jerk for Trajectory Planning and Control, Robotica, 12/2:109-113.

[35] Butler, J., Haack, B., Tomizuka, M., 1988, Reference Generation for High Speed Coordinated Motion of a Two Axis System, Symposium on Robotics, Winter Annual Meeting of the American Society of Mechanical Engineers, DSC-Vol. 11, pp. 457-470.

[36] Erdim, H., Lazoglu, I., Ozturk, B., 2006, Feedrate scheduling strategies for free-form surfaces, International Journal of Machine Tools and Manufacture, 46/7-8:747-757.

[37] Erkorkmaz, K., Heng, M., 2008, A Heuristic Feedrate Optimization Strategy for NURBS Toolpaths, To appear in: Annals of CIRP, 57/1.

[38] Mathews, J.H., Fink, K.K., 2004, *Numerical methods using Matlab, 4th Edition*; Prentice-Hall Inc, Upper Saddle River, New Jersey, pp. 391-395.

[39] Erkorkmaz, K., Altintas, Y., 2001, High speed CNC system design. Part II: modelling and identification of feed drives, International Journal of Machine Tools and Manufacture, 41/8:1487-1509.

[40] Altintas, Y., Erkorkmaz, K., Zhu, W.-H., 2000, Sliding Mode Controller Design for High Speed Feed Drives, Annals of the CIRP, 49/1:265-270.

[41] Heng, M., Erkorkmaz, K., 2007, Feedrate Optimization of Spline Toolpaths on Machine Tools with Dynamic Uncertainty, Proceedings of the 21st Canadian Congress of Applied Mechanics, June 3-7, 2007.

Appendix A

Non Uniform Rational B-Spline (NURBS) Format

The data points used to generate the NURBS fan-shaped toolpath are from [8] scaled by 150%. There are 89 data points.

Fan-shaped Toolpath Data Points

$\mathbf{q}_x$ = [-16.0688, -0.0562, 7.8712, 13.5113, 15.0488, 13.5863, 9.1125, 8.2125, 8.4150, 9.2025, 10.5413, 12.0825, 13.5825, 15.1650, 18.1950, 25.5900, 31.6463, 34.3387, 37.6013, 41.3325, 45.1275, 49.6200, 56.9850, 57.1575, 54.1838, 48.1388, 39.2062, 30.1613, 15.1537, 11.8650, 10.4175, 9.0300, 8.3963, 8.3438, 8.5313, 9.0975, 10.6463, 18.3188, 27.3038, 30.2400, 32.4675, 33.5887, 33.2400, 30.2137, 16.0688, 0.0562, -7.8712, -13.5113, -15.0488, -13.5863, -9.1125, -8.2125, -8.4150, -9.2025, -10.5413, -12.0825, -13.5825, -15.1650, -18.1950, -25.5900, -31.6463, -34.3387, -37.6013, -41.3325, -45.1275, -49.6200, -56.9850, -57.1575, -54.1838, -48.1388, -39.2062, -30.1613, -15.1537, -11.8650, -10.4175, -9.0300, -8.3963, -8.3438, -8.5313, -9.0975, -10.6463, -18.3188, -27.3038, -30.2400, -32.4675, -33.5887, -33.2400, -30.2137, -16.0688]

$\mathbf{q}_y$ = [-56.9850, -57.1575, -54.1838, -48.1388, -39.2062, -30.1613, -15.1537, - 11.8650, -10.4175, -9.0300, -8.3963, -8.3438, -8.5313, -9.0975, -10.6463, -18.3188, -27.3038, -30.2400, -32.4675, -33.5887, -33.2400, -30.2137, -16.0688, -0.0562, 7.8712, 13.5113, 15.0488, 13.5863, 9.1125, 8.2125, 8.4150, 9.2025, 10.5413, 12.0825, 13.5825, 15.1650, 18.1950, 25.5900, 31.6463, 34.3387, 37.6013, 41.3325, 45.1275, 49.6200, 56.9850, 57.1575, 54.1838, 48.1388, 39.2062, 30.1613, 15.1537, 11.8650, 10.4175, 9.0300, 8.3963, 8.3438, 8.5313, 9.0975, 10.6463, 18.3188, 27.3038, 30.2400, 32.4675, 33.5887, 3.2400, 30.2137, 16.0688, 0.0562, -7.8712, -13.5113, -15.0488, -13.5863, -9.1125, -8.2125, -8.4150, -9.2025, -10.5413, -12.0825, -13.5825, -15.1650, -18.1950, -25.5900, -31.6463, -34.3387, -37.6013, -41.3325, -45.1275, -49.6200, -56.9850]

The beta shape parameters, β1 and β2 , used in the NURBS toolpath parameterization of the fan-shaped toolpath are listed here. Note that a start (*) means that the beta value β1 was manually modified to reduce oscillations in the curve.

Segment, *k*	β1	β2
1	0.4516	0
2	0.7761	0
3	1.2829	0
4*	1.0000	0
5	0.6761	0
6*	0.4000	0
7*	2.0000	0
8	1.7025	0
9	1.8791	0
10*	0.4000	0
11	2.4056	0
12*	1.0000	0
13*	1.5000	0
14	0.4544	0
15	0.3781	0
16*	4.0000	0
17	0.7434	0

The NURBS format consists of the degree *p*, the number of control points $n + 1$, a knot vector **U**, control points $\mathbf{P}_i$, and weights w_i. The NURBS parameterization generated 17 segment each with degree of three and six control points, except the 17th segment, which has nine control points. All weights are equal to one.

In the following table, the knot vector and control points are listed sequentially for each segment. Only the unique values of the knot vectors are listed, however the full knot vector would include 3 more zeros at the beginning and 3 more zeros at the end. Since the first control point of a segment is the same as the last control point of the previous segment, it is omitted in the entries for all but the first segment.

Segment	Knots	Control Points (x,y)
k	$u_{1,p+1}, \ldots , u_{1,m-p}$	$\mathbf{P}_{k,0}, \ldots, \mathbf{P}_{k,n}$
1	0, 0.4789, 0.6476, 1	(-16.0694, -56.9551) (-8.3336, -58.5969) (4.1621, -58.7056) (16.7467, -45.7588) (15.1264, -36.0089) (13.5246, -30.0387)

k	$u_{k,p+1}, \ldots, u_{k,m-p}$	$\mathbf{P}_{k,1}, \ldots, \mathbf{P}_{k,n}$
2	0, 0.7803, 0.8716, 1	(11.9228, -24.0685) (9.5683, -18.0956) (7.9234, -9.9527) (9.5652, -8.6741) (10.5648, -8.3924)
3	0, 0.1655, 0.2825, 1	(11.5644, -8.1106) (13.3390, -8.4241) (19.3847, -10.6097) (22.8611, -14.3091) (25.5827, -18.3011)

k	$u_{k,p+1}, \ldots, u_{k,m-p}$	$\mathbf{P}_{k,1}, \ldots, \mathbf{P}_{k,n}$
4	0, 0.5593, 0.7082, 1	(28.3042, -22.2932) (31.4117, -28.3756) (38.3488, -33.0681) (42.5078, -34.7395) (45.6878, -33.0119)
5	0, 0.3952, 0.6443, 1	(49.9954, -30.6717) (58.2932, -18.3800) (58.6229, -1.0776) (53.8510, 10.7472) (48.1719, 13.4772)
6	0, 0.5261, 0.7690, 1	(42.4928, 16.2073) (29.9952, 14.0217) (21.1425, 11.0262) (12.6160, 8.0204) (10.5200, 8.3398)
7	0, 0.3965, 0.5900, 1	(9.0812, 8.5591) (8.3854, 10.1370) (8.3008, 12.4714) (8.4844, 14.1095) (9.1682, 15.5321)
8	0, 0.4301, 0.6887, 1	(10.6028, 18.5165) (16.8407, 25.4791) (25.8525, 29.9687) (31.0562, 34.7591) (32.4800, 37.5026)
9	0, 0.1828, 0.3690, 1	(33.9038, 40.2461) (33.9155, 46.0398) (23.2687, 57.4627) (8.2387, 58.5978) (0.4939, 57.3339)
10	0, 0.3358, 0.5103, 1	(-7.2510, 56.0700) (-13.1514, 50.8971) (-16.7405, 34.6472) (-10.8161, 20.7484) (-9.0215, 15.4369)

k	$u_{k,p+1}, \ldots, u_{k,m-p}$	$\mathbf{P}_{k,1}, \ldots, \mathbf{P}_{k,n}$
11	0, 0.5180, 0.6774, 1	(-8.2622, 13.1895) (-7.8718, 11.4062) (-9.3412, 8.5122) (-10.9515, 8.1901) (-12.1249, 8.2972)
12	0, 0.1341, 0.3209, 1	(-13.2983, 8.4044) (-16.3351, 9.4330) (-23.8354, 13.9873) (-28.3362, 22.9256) (-31.6135, 27.2606)
13	0, 0.3759, 0.5604, 1	(-33.4276, 29.6603) (-36.0170, 32.0157) (-43.2263, 34.5107) (-47.1266, 33.0694) (-49.8786, 29.8837)
14	0, 0.5098, 0.6988, 1	(-54.6663, 24.3416) (-61.2492, 7.7404) (-53.8494, -10.4212) (-45.7185, -15.4743) (-39.2635, -15.2976)
15	0, 0.6628, 0.8815, 1	(-32.8085, -15.1208) (-22.2832, -10.5213) (-13.8906, -8.5210) (-9.8434, -8.3416) (-9.0839, -9.0280)
16	0, 0.3133, 0.4773, 1	(-8.3243, -9.7144) (-8.1762, -11.9022) (-8.7961, -14.5951) (-9.5323, -16.6295) (-10.7083, -18.2904)

k	$u_{k,p+1}$, ..., $u_{k,m-p}$	$\mathbf{P}_{k,1}$, ..., $\mathbf{P}_{k,n}$
17	0, 0.3284, 0.4353, 0.5027, 0.5691, 0.6439, 1	(-13.6634, -22.4642) (-22.7358, -28.9401) (-29.7394, -33.1950) (-32.5702, -37.6368) (-34.2249, -41.3847) (-31.3445, -51.4373) (-23.8052, -55.3133) (-16.0694, -56.9551)

Appendix B

Kinematic Compatibility Derivations

In this appendix, expressions for the time duration of each phase (constant jerk, constant acceleration, and constant feedrate) are derived for an *N*-segment toolpath. Using these expressions, the kinematic compatibility conditions are also derived.

Constant Jerk Phase Time Duration

From the trapezoidal nature of the acceleration profile, the acceleration value reached in the time duration $T_{j,k}$ $(1 \le k \le N+1)$ is:

$$A_k = J_k T_{j,k} \rightarrow T_{j,k} = \frac{A_k}{J_k} \tag{B.1}$$

Constant Acceleration Phase Time Duration

Considering that the desired feed of the initial segment (F_1) must be reached by the end of the first acceleration transient, let $f_{3e} = F_1$. From Equation (4.8),

$$\begin{aligned} f_s &= 0 \\ f_{1e} &= f_s + \frac{1}{2} J_1 T_{j,1}^2 \\ f_{2e} &= f_{1e} + A_1 T_{a,1} \\ f_{3e} &= f_{2e} + A_1 T_{j,1} - \frac{1}{2} J_1 T_{j,1}^2 \end{aligned} \tag{B.2}$$

To obtain an explicit expression for f_{3e}, substitute the expression for the end feed of the previous phase into the next equation recursively. That is:

$$\begin{aligned} f_s &= 0 \\ f_{1e} &= 0 + \frac{1}{2} J_1 T_{j,1}^2 \\ f_{2e} &= \frac{1}{2} J_1 T_{j,1}^2 + A_1 T_{a,1} \\ f_{3e} &= F_1 = \frac{1}{2} J_1 T_{j,1}^2 + A_1 T_{a,1} + A_1 T_{j,1} - \frac{1}{2} J_1 T_{j,1}^2 \end{aligned} \tag{B.3}$$

Therefore after simplification:

$$F_1 = A_1 T_{a,1} + A_1 T_{j,1} \tag{B.4}$$

Substitute Equation (B.1) into Equation (B.4) to get:

$$F_1 = A_1 T_{a,1} + \frac{{A_1}^2}{J_1} \tag{B.5}$$

Solving for the constant acceleration time duration of the initial segment yields:

$$T_{a,1} = \frac{F_1}{A_1} - \frac{A_1}{J_1} \tag{B.6}$$

Now solving for the *k*th constant acceleration time duration $T_{a,k}$, the desired feed (F_k) must be reached by the end of the *k*th acceleration transient. Thus, let $f_{2e} = F_k$. From Equation (4.9),

$$\begin{aligned} f_s &= (F_{k-1} + F_k)/2 \\ f_{1e} &= f_s + \frac{1}{2} A_k T_{a,k} \\ f_{2e} &= f_{1e} + A_k T_{j,k} - \frac{1}{2} J_k T_{j,k}^2 \end{aligned} \tag{B.7}$$

Then by recursively substituting the previous expression into the next gives:

$$\begin{aligned}
f_s &= (F_{k-1}+F_k)/2 \\
f_{1e} &= (F_{k-1}+F_k)/2+\frac{1}{2}A_k T_{a,k} \\
f_{2e} = F_k &= (F_{k-1}+F_k)/2+\frac{1}{2}A_k T_{a,k}+A_k T_{j,k}-\frac{1}{2}J_k T_{j,k}^2
\end{aligned} \tag{B.8}$$

Substitute Equation (B.1) into Equation (B.8) and simplify.

$$F_k = (F_{k-1}+F_k)/2+\frac{1}{2}A_k T_{a,k}+A_k\frac{A_k}{J_k}-\frac{1}{2}J_k\left(\frac{A_k}{J_k}\right)^2$$

$$\begin{aligned}
F_k-(F_{k-1}+F_k)/2 &= \frac{1}{2}A_k T_{a,k}+\frac{1}{2}\frac{A_k^2}{J_k} \\
(F_k-F_{k-1})/2 &= \frac{1}{2}A_k T_{a,k}+\frac{1}{2}\frac{A_k^2}{J_k} \\
T_{a,k} &= \frac{F_k-F_{k-1}}{A_k}-\frac{A_k}{J_k}
\end{aligned} \tag{B.9}$$

Equation (B.9) is also generalizable to the initial segment $(k=1)$ by setting $F_0=0$.

Constant Feedrate Phase Time Duration

Initial (1st) Segment

Considering that the total travel length by the end of the motion must be equal to the segment arc-length (L_k), the constant feedrate time duration $(T_{f,k})$ can be obtained by equating the segment arc-length to the total length traveled. A different expression results for each type of segment. To start, let us consider the initial segment and define s_{6e} as the total length traveled. From Equations (4.8) and (4.11),

$$
\begin{array}{ll}
\begin{aligned}
f_s &= 0 \\
f_{1e} &= f_s + \frac{1}{2} J_1 T_{j,1}^2 \\
f_{2e} &= f_{1e} + A_1 T_{a,1} \\
f_{3e} &= f_{2e} + A_1 T_{j,1} - \frac{1}{2} J_1 T_{j,1}^2 \\
f_{4e} &= f_{3e} \\
f_{5e} &= f_{4e} + \frac{1}{2} J_2 T_{j,2}^2
\end{aligned}
&
\begin{aligned}
s_s &= 0 \\
s_{1e} &= s_s + f_s T_{j,1} + \frac{1}{6} J_1 T_{j,1}^3 \\
s_{2e} &= s_{1e} + f_{1e} T_{a,1} + \frac{1}{2} A_1 T_{a,1}^2 \\
s_{3e} &= s_{2e} + f_{2e} T_{j,1} + \frac{1}{2} A_1 T_{j,1}^2 - \frac{1}{6} J_1 T_{j,1}^3 \\
s_{4e} &= s_{3e} + f_{3e} T_{f,1} \\
s_{5e} &= s_{4e} + f_{4e} T_{j,2} + \frac{1}{6} J_2 T_{j,2}^3 \\
s_{6e} &= s_{5e} + f_{5e} \frac{T_{a,2}}{2} + \frac{1}{8} A_2 T_{a,2}^2
\end{aligned}
\end{array}
\tag{B.10}
$$

Also, recall that the feed reached at the end of the third phase is equal to the desired feedrate, i.e. $f_{3e} = F_1$.

To obtain an explicit equation for s_{6e}, expressions for the end feed are substituted into the equations for the length traveled by the end of each phase.

$$
\begin{aligned}
f_s &= F_0 = 0 \\
f_{1e} &= F_0 + \frac{1}{2} J_1 T_{j,1}^2 \\
f_{2e} &= F_0 + \frac{1}{2} J_1 T_{j,1}^2 + A_1 T_{a,1} \quad \rightarrow \\
f_{3e} &= F_1 \\
f_{4e} &= F_1 \\
f_{5e} &= F_1 + \frac{1}{2} J_2 T_{j,2}^2
\end{aligned}
\tag{B.11}
$$

$$
\begin{aligned}
s_s &= 0 \\
s_{1e} &= s_s + F_0 T_{j,1} + \frac{1}{6} J_1 T_{j,1}^3 \\
s_{2e} &= s_{1e} + (F_0 + \frac{1}{2} J_1 T_{j,1}^2) T_{a,1} + \frac{1}{2} A_1 T_{a,1}^2 \\
s_{3e} &= s_{2e} + (F_0 + \frac{1}{2} J_1 T_{j,1}^2 + A_1 T_{a,1}) T_{j,1} + \frac{1}{2} A_1 T_{j,1}^2 - \frac{1}{6} J_1 T_{j,1}^3 \\
s_{4e} &= s_{3e} + F_1 T_{f,1} \\
s_{5e} &= s_{4e} + F_1 T_{j,2} + \frac{1}{6} J_2 T_{j,2}^3 \\
s_{6e} &= s_{5e} + (F_1 + \frac{1}{2} J_2 T_{j,2}^2) \frac{T_{a,2}}{2} + \frac{1}{8} A_2 T_{a,2}^2
\end{aligned}
$$

Then expressions for the length traveled in the previous phase are substituted into the next phase equations recursively to yield:

$$
\begin{aligned}
s_{6e} = {} & F_0 T_{j,1} + \frac{1}{6} J_1 T_{j,1}^3 + (F_0 + \frac{1}{2} J_1 T_{j,1}^2) T_{a,1} + \frac{1}{2} A_1 T_{a,1}^2 + (F_0 + \frac{1}{2} J_1 T_{j,1}^2 + A_1 T_{a,1}) T_{j,1} \\
& + \frac{1}{2} A_1 T_{j,1}^2 - \frac{1}{6} J_1 T_{j,1}^3 + F_1 T_{f,1} + F_1 T_{j,2} + \frac{1}{6} J_2 T_{j,2}^3 + (F_1 + \frac{1}{2} J_2 T_{j,2}^2) \frac{T_{a,2}}{2} + \frac{1}{8} A_2 T_{a,2}^2
\end{aligned}
\tag{B.12}
$$

Substituting in $J_k = A_k / T_{j,k}$ and grouping terms in Equation (B.12) results in the following equation:

$$
\begin{aligned}
s_{6e} = {} & F_0 \left[2T_{j,1} + T_{a,1} \right] + A_1 \left[T_{j,1}^2 + \frac{3}{2} T_{j,1} T_{a,1} + \frac{1}{2} T_{a,1}^2 \right] \\
& + F_1 \left[T_{f,1} + T_{j,2} + \frac{T_{a,2}}{2} \right] + A_2 \left[\frac{1}{6} T_{j,2}^2 + \frac{1}{4} T_{j,2} T_{a,2} + \frac{1}{8} T_{a,2}^2 \right]
\end{aligned}
\tag{B.13}
$$

Substitute $T_{j,k}$, $T_{a,1}$, and $T_{a,k}$, Equations (B.1), (B.6) and (B.9), respectively, into Equation (B.13).

$$\begin{aligned} s_{6e} &= F_0\left[\frac{2A_1}{J_1}+\frac{F_1-F_0}{A_1}-\frac{A_1}{J_1}\right] \\ &+A_1\left[\frac{{A_1}^2}{{J_1}^2}+\frac{3}{2}\frac{A_1}{J_1}\left(\frac{F_1-F_0}{A_1}-\frac{A_1}{J_1}\right)+\frac{1}{2}\left(\frac{F_1-F_0}{A_1}-\frac{A_1}{J_1}\right)^2\right] \\ &+F_1\left[T_{f,1}+\frac{A_2}{J_2}+\frac{F_2-F_1}{2A_2}-\frac{A_2}{2J_2}\right] \\ &+A_2\left[\frac{1}{6}\frac{{A_2}^2}{{J_2}^2}+\frac{1}{4}\frac{A_2}{J_2}\left(\frac{F_2-F_1}{A_2}-\frac{A_2}{J_2}\right)+\frac{1}{8}\left(\frac{F_2-F_1}{A_2}-\frac{A_2}{J_2}\right)^2\right] \end{aligned} \tag{B.14}$$

Expand and simplify Equation (B.14).

$$\begin{aligned} s_{6e} &= \frac{F_0A_1}{J_1}+\frac{F_0(F_1-F_0)}{A_1} \\ &+\frac{A_1^3}{J_1^2}+\frac{3}{2}\frac{A_1(F_1-F_0)}{J_1}-\frac{3}{2}\frac{A_1^3}{J_1^2}+\frac{(F_1-F_0)^2}{2A_1}-\frac{2A_1(F_1-F_0)}{2J_1}+\frac{A_1^3}{2J_1^2} \\ &+F_1T_{f,1}+\frac{F_1A_2}{J_2}+\frac{F_1(F_2-F_1)}{2A_2}-\frac{F_1A_2}{2J_2} \\ &+\frac{1}{6}\frac{A_2^3}{J_2^2}+\frac{1}{4}\frac{A_2(F_2-F_1)}{J_2}-\frac{A_2^3}{4J_2^2}+\frac{(F_2-F_1)^2}{8A_2}-\frac{2A_2(F_2-F_1)}{8J_2}+\frac{A_2^3}{8J_2^2} \end{aligned} \tag{B.15}$$

Thus,

$$\begin{aligned} s_{6e} &= \frac{F_0A_1}{J_1}+\frac{F_0(F_1-F_0)}{A_1}+\frac{A_1(F_1-F_0)}{2J_1}+\frac{(F_1-F_0)^2}{2A_1} \\ &+F_1T_{f,1}+\frac{F_1A_2}{2J_2}+\frac{A_2^3}{24J_2^2}+\frac{F_1(F_2-F_1)}{2A_2}+\frac{(F_2-F_1)^2}{8A_2} \end{aligned} \tag{B.16}$$

Define Δ_k as $\Delta_k = F_k - F_{k-1}$. Then, $F_{k-1} = F_k - \Delta_k$. Substituting the former relation into Equation (B.16) yields,

$$s_{6e} = \frac{F_0A_1}{J_1}+\frac{F_0\Delta_1}{A_1}+\frac{A_1\Delta_1}{2J_1}+\frac{{\Delta_1}^2}{2A_1}+F_1T_{f,1}+\frac{F_1A_2}{2J_2}+\frac{A_2^3}{24J_2^2}+\frac{\overbrace{F_1}^{*}\Delta_2}{2A_2}+\frac{{\Delta_2}^2}{8A_2} \tag{B.17}$$

Here we will substitute $F_1 = F_2 - \Delta_2$ into the starred (*) term of Equation (B.17).

$$\begin{gathered}
s_{6e} = \frac{F_0 A_1}{J_1} + \frac{F_0 \Delta_1}{A_1} + \frac{A_1 \Delta_1}{2J_1} + \frac{{\Delta_1}^2}{2A_1} + F_1 T_{f,1} + \frac{F_1 A_2}{2J_2} + \frac{A_2^3}{24J_2^2} + \frac{\overbrace{(F_2 - \Delta_2)}^{*}\Delta_2}{2A_2} + \frac{{\Delta_2}^2}{8A_2} \\
\downarrow \\
s_{6e} = \frac{F_0 A_1}{J_1} + \frac{F_0 \Delta_1}{A_1} + \frac{A_1 \Delta_1}{2J_1} + \frac{{\Delta_1}^2}{2A_1} + F_1 T_{f,1} + \frac{F_1 A_2}{2J_2} + \frac{A_2^3}{24J_2^2} + \frac{F_2 \Delta_2}{2A_2} - \frac{3{\Delta_2}^2}{8A_2}
\end{gathered} \tag{B.18}$$

Then letting $s_{6e} = L_1$ and solving for $T_{f,1}$ results in the following expression for the constant feedrate time duration:

$$T_{f,1} = \frac{L_1}{F_1} - \frac{1}{F_1}\left[\frac{F_0 A_1}{J_1} + \frac{F_0 \Delta_1}{A_1} + \frac{{\Delta_1}^2}{2A_1} + \frac{A_1 \Delta_1}{2J_1} - \frac{3{\Delta_2}^2}{8A_2} + \frac{A_2^3}{24J_2^2} + \frac{F_1 A_2}{2J_2} + \frac{F_2 \Delta_2}{2A_2}\right] \tag{B.19}$$

Note that by definition of the initial segment, the start feed F_0 is equal to zero. It is still included here for completeness.

Middle (kth) Segment

Considering the *k*th $(2 \le k \le N-1)$ constant feed phase, s_{5e} is the total distance traveled. From Equations (4.9) and (4.12), the feeds and travel lengths at the end of each phase are listed below:

$$\begin{aligned}
f_s &= (F_{k-1} + F_k)/2 \\
f_{1e} &= f_s + \frac{1}{2} A_k T_{a,k} \\
f_{2e} &= f_{1e} + A_k T_{j,k} - \frac{1}{2} J_k T_{j,k}^2 \\
f_{3e} &= f_{2e} \\
f_{4e} &= f_{3e} + \frac{1}{2} J_{k+1} T_{j,k+1}^2
\end{aligned}
\qquad
\begin{aligned}
s_s &= 0 \\
s_{1e} &= s_s + \frac{1}{2} f_s T_{a,k} + \frac{1}{8} A_k T_{a,k}^2 \\
s_{2e} &= s_{1e} + f_{1e} T_{j,k} + \frac{1}{2} A_k T_{j,k}^2 - \frac{1}{6} J_k T_{j,k}^3 \\
s_{3e} &= s_{2e} + f_{2e} T_{f,k} \\
s_{4e} &= s_{3e} + f_{3e} T_{j,k+1} + \frac{1}{6} J_{k+1} T_{j,k+1}^3 \\
s_{5e} &= s_{4e} + \frac{1}{2} f_{4e} T_{a,k+1} + \frac{1}{8} A_{k+1} T_{a,k+1}^2
\end{aligned} \tag{B.20}$$

Recall that the desired feedrate (F_k) is reached by the end of the second phase for mid-segments, that is $f_{2e} = F_k$. Recursively solving for the end feeds of each phase and substituting them into the travel length equations results in the following algebraic steps:

$$\begin{array}{lcl} \begin{aligned} f_s &= (F_{k-1} + F_k)/2 \\ f_{1e} &= (F_{k-1} + F_k)/2 + \frac{1}{2} A_k T_{a,k} \\ f_{2e} &= F_k \\ f_{3e} &= F_k \\ f_{4e} &= F_k + \frac{1}{2} J_{k+1} T_{j,k+1}^2 \end{aligned} & \rightarrow & \begin{aligned} s_s &= 0 \\ s_{1e} &= s_s + \frac{1}{4}(F_{k-1} + F_k) T_{a,k} + \frac{1}{8} A_k T_{a,k}^2 \\ s_{2e} &= s_{1e} + (\frac{F_{k-1} + F_k}{2} + \frac{1}{2} A_k T_{a,k}) T_{j,k} \\ &\quad + \frac{1}{2} A_k T_{j,k}^2 - \frac{1}{6} J_k T_{j,k}^3 \\ s_{3e} &= s_{2e} + F_k T_{f,k} \\ s_{4e} &= s_{3e} + F_k T_{j,k+1} + \frac{1}{6} J_{k+1} T_{j,k+1}^3 \\ s_{5e} &= s_{4e} + \frac{1}{2}(F_k + \frac{1}{2} J_{k+1} T_{j,k+1}^2) T_{a,k+1} \\ &\quad + \frac{1}{8} A_{k+1} T_{a,k+1}^2 \end{aligned} \end{array} \qquad \text{(B.21)}$$

To obtain an explicit equation for s_{5e}, travel length equations are recursively substituted into the expression for s_{5e} which results in the following equation:

$$\begin{aligned} s_{5e} &= \frac{1}{4}(F_{k-1} + F_k) T_{a,k} + \frac{1}{8} A_k T_{a,k}^2 + (\frac{F_{k-1} + F_k}{2} + \frac{1}{2} A_k T_{a,k}) T_{j,k} + \frac{1}{2} A_k T_{j,k}^2 - \frac{1}{6} J_k T_{j,k}^3 \\ &+ F_k T_{f,k} + F_k T_{j,k+1} + \frac{1}{6} J_{k+1} T_{j,k+1}^3 + \frac{1}{2}(F_k + \frac{1}{2} J_{k+1} T_{j,k+1}^2) T_{a,k+1} + \frac{1}{8} A_{k+1} T_{a,k+1}^2 \end{aligned} \qquad \text{(B.22)}$$

Substituting in $J_k = A_k / T_{j,k}$ and grouping terms in Equation (B.22) results in the following equation:

$$\begin{aligned} s_{5e} &= F_{k-1}\left[\frac{1}{2}T_{j,k} + \frac{1}{4}T_{a,k}\right] \\ &+ A_k\left[\frac{1}{3}T_{j,k}^2 + \frac{1}{2}T_{j,k}T_{a,k} + \frac{1}{8}T_{a,k}^2\right] \\ &+ F_k\left[\frac{1}{2}T_{j,k} + \frac{1}{4}T_{a,k} + T_{f,k} + T_{j,k+1} + \frac{1}{2}T_{a,k+1}\right] \\ &+ A_{k+1}\left[\frac{1}{6}T_{j,k+1}^2 + \frac{1}{4}T_{j,k+1}T_{a,k+1} + \frac{1}{8}T_{a,k+1}^2\right] \end{aligned} \tag{B.23}$$

Substitute $T_{j,k}$ and $T_{a,k}$, Equations (B.1) and (B.9), respectively, into Equation (B.23).

$$\begin{aligned} s_{5e} &= F_{k-1}\left[\frac{1}{2}\frac{A_k}{J_k} + \frac{1}{4}\left(\frac{F_k - F_{k-1}}{A_k} - \frac{A_k}{J_k}\right)\right] \\ &+ A_k\left[\frac{1}{3}\frac{A_k^2}{J_k^2} + \frac{1}{2}\frac{A_k}{J_k}\left(\frac{F_k - F_{k-1}}{A_k} - \frac{A_k}{J_k}\right) + \frac{1}{8}\left(\frac{F_k - F_{k-1}}{A_k} - \frac{A_k}{J_k}\right)^2\right] \\ &+ F_k\left[\frac{1}{2}\frac{A_k}{J_k} + \frac{1}{4}\left(\frac{F_k - F_{k-1}}{A_k} - \frac{A_k}{J_k}\right) + T_{f,k} + \frac{A_{k+1}}{J_{k+1}} + \frac{1}{2}\left(\frac{F_{k+1} - F_k}{A_{k+1}} - \frac{A_{k+1}}{J_{k+1}}\right)\right] \\ &+ A_{k+1}\left[\frac{1}{6}\frac{A_{k+1}^2}{J_{k+1}^2} + \frac{1}{4}\frac{A_{k+1}}{J_{k+1}}\left(\frac{F_{k+1} - F_k}{A_{k+1}} - \frac{A_{k+1}}{J_{k+1}}\right) + \frac{1}{8}\left(\frac{F_{k+1} - F_k}{A_{k+1}} - \frac{A_{k+1}}{J_{k+1}}\right)^2\right] \end{aligned} \tag{B.24}$$

Expand and simplify Equation (B.24).

$$\begin{aligned} s_{5e} &= \frac{F_{k-1}(F_k - F_{k-1})}{4A_k} - \frac{F_{k-1}A_k}{4J_k} \\ &+ \frac{A_k^3}{3J_k^2} + \frac{(F_k - F_{k-1})A_k}{2J_k} - \frac{A_k^3}{2J_k^2} + \frac{(F_k - F_{k-1})^2}{8A_k} - \frac{2(F_k - F_{k-1})A_k}{8J_k} + \frac{A_k^3}{8J_k^2} \\ &+ \frac{F_k A_k}{2J_k} + \frac{F_k(F_k - F_{k-1})}{4A_k} - \frac{F_k A_k}{4J_k} + F_k T_{f,k} + \frac{F_k A_{k+1}}{J_{k+1}} + \frac{F_k(F_{k+1} - F_k)}{2A_{k+1}} - \frac{F_k A_{k+1}}{2J_{k+1}} \\ &+ \frac{A_{k+1}^3}{6J_{k+1}^2} + \frac{(F_{k+1} - F_k)A_{k+1}}{4J_{k+1}} - \frac{A_{k+1}^3}{4J_{k+1}^2} + \frac{(F_{k+1} - F_k)^2}{8A_{k+1}} - \frac{2(F_{k+1} - F_k)A_{k+1}}{8J_{k+1}} + \frac{A_{k+1}^3}{8J_{k+1}^2} \end{aligned} \tag{B.25}$$

Thus,

$$\begin{aligned} s_{5e} = {} & \frac{F_{k-1}(F_k - F_{k-1})}{4A_k} + \frac{F_k(F_k - F_{k-1})}{4A_k} + \frac{(F_k - F_{k-1})^2}{8A_k} + \frac{F_k A_k}{2J_k} - \frac{A_k^3}{24J_k^2} \\ & + F_k T_{f,k} + \frac{F_k A_{k+1}}{2J_{k+1}} + \frac{A_{k+1}^3}{24J_{k+1}^2} + \frac{F_k(F_{k+1} - F_k)}{2A_{k+1}} + \frac{(F_{k+1} - F_k)^2}{8A_{k+1}} \end{aligned} \tag{B.26}$$

Substituting the $\Delta_k = F_k - F_{k-1}$ into Equation (B.26) yields:

$$\begin{aligned} s_{5e} = {} & \frac{F_{k-1}\Delta_k}{4A_k} + \frac{\overbrace{F_k}^{*1}\Delta_k}{4A_k} + \frac{{\Delta_k}^2}{8A_k} + \frac{F_k A_k}{2J_k} - \frac{A_k^3}{24J_k^2} + F_k T_{f,k} + \\ & \frac{F_k A_{k+1}}{2J_{k+1}} + \frac{A_{k+1}^3}{24J_{k+1}^2} + \frac{\overbrace{F_k}^{*2}\Delta_{k+1}}{2A_{k+1}} + \frac{{\Delta_{k+1}}^2}{8A_{k+1}} \end{aligned} \tag{B.27}$$

Here we will substitute $F_k = F_{k-1} + \Delta_k$ into the first starred term (*1) and $F_k = F_{k+1} - \Delta_{k+1}$ into the second starred term (*2) of Equation (B.27).

$$\begin{aligned} s_{5e} = {} & \frac{F_{k-1}\Delta_k}{4A_k} + \frac{\overbrace{(F_{k-1} + \Delta_k)}^{*1}\Delta_k}{4A_k} + \frac{{\Delta_k}^2}{8A_k} + \frac{F_k A_k}{2J_k} - \frac{A_k^3}{24J_k^2} + F_k T_{f,k} \\ & + \frac{F_k A_{k+1}}{2J_{k+1}} + \frac{A_{k+1}^3}{24J_{k+1}^2} + \frac{\overbrace{(F_{k+1} - \Delta_{k+1})}^{*2}\Delta_{k+1}}{2A_{k+1}} + \frac{{\Delta_{k+1}}^2}{8A_{k+1}} \\ & \downarrow \\ s_{5e} = {} & \frac{F_{k-1}\Delta_k}{2A_k} + \frac{3{\Delta_k}^2}{8A_k} + \frac{F_k A_k}{2J_k} - \frac{A_k^3}{24J_k^2} + F_k T_{f,k} \\ & + \frac{F_k A_{k+1}}{2J_{k+1}} + \frac{A_{k+1}^3}{24J_{k+1}^2} + \frac{F_{k+1}\Delta_{k+1}}{2A_{k+1}} - \frac{3{\Delta_{k+1}}^2}{8A_{k+1}} \end{aligned} \tag{B.28}$$

Then letting $s_{5e} = L_k$ and solving for $T_{f,k}$ results in the following expression for the constant feedrate time duration for the *k*th segment $(2 \le k \le N-1)$:

$$T_{f,k}=\frac{L_k}{F_k}-\frac{1}{F_k}\left[\begin{array}{l}\dfrac{3\Delta_k{}^2}{8A_k}-\dfrac{A_k^3}{24J_k^2}+\dfrac{F_kA_k}{2J_k}+\dfrac{F_{k-1}\Delta_k}{2A_k}\\ -\dfrac{3\Delta_{k+1}{}^2}{8A_{k+1}}+\dfrac{A_{k+1}^3}{24J_{k+1}^2}+\dfrac{F_kA_{k+1}}{2J_{k+1}}+\dfrac{F_{k+1}\Delta_{k+1}}{2A_{k+1}}\end{array}\right] \tag{B.29}$$

Final (Nth) Segment

Lastly, considering the final segment $(k = N)$, the constant feed time duration is found similarly. The total travel distance of the final segment is denoted as s_{6e}. From Equations (4.10) and (4.13), the feed and distance traveled at the end of each phase are listed below:

$$\begin{aligned}
f_s &= (F_{N-1}+F_N)/2\\
f_{1e} &= f_s+\frac{1}{2}A_NT_{a,N}\\
f_{2e} &= f_{1e}+A_NT_{j,N}-\frac{1}{2}J_NT_{j,N}^2\\
f_{3e} &= f_{2e}\\
f_{4e} &= f_{3e}+\frac{1}{2}J_{N+1}T_{j,N+1}^2\\
f_{5e} &= f_{4e}+A_{N+1}T_{a,N+1}\\
s_s &= 0\\
s_{1e} &= s_s+\frac{1}{2}f_sT_{a,N}+\frac{1}{8}A_NT_{a,N}^2\\
s_{2e} &= s_{1e}+f_{1e}T_{j,N}+\frac{1}{2}A_NT_{j,N}^2-\frac{1}{6}J_NT_{j,N}^3\\
s_{3e} &= s_{2e}+f_{2e}T_{f,N}\\
s_{4e} &= s_{3e}+f_{3e}T_{j,N+1}+\frac{1}{6}J_{N+1}T_{j,N+1}^3\\
s_{5e} &= s_{4e}+f_{4e}T_{a,N+1}+\frac{1}{2}A_{N+1}T_{a,N}^2\\
s_{6e} &= s_{5e}+f_{5e}T_{j,N+1}+\frac{1}{2}A_{N+1}T_{j,N+1}^2-\frac{1}{6}J_{N+1}T_{j,N+1}^3
\end{aligned} \tag{B.30}$$

The desired feedrate of the final segment (F_N) is reached by the end of the second phase as in the mid-segment, that is $f_{2e} = F_N$. Recursively solving for the end feeds of each phase and substituting them into the travel length equations results in the following equations:

$$\begin{aligned}
f_s &= \left(F_{N-1}+F_N\right)/2 \\
f_{1e} &= \left(F_{N-1}+F_N\right)/2+\frac{1}{2}A_N T_{a,N} \\
f_{2e} &= F_N \\
f_{3e} &= F_N \\
f_{4e} &= F_N+\frac{1}{2}J_{N+1}T_{j,N+1}^2 \\
f_{5e} &= F_N+\frac{1}{2}J_{N+1}T_{j,N+1}^2+A_{N+1}T_{a,N+1}
\end{aligned}$$

$$\downarrow$$

$$\begin{aligned}
s_s &= 0 \\
s_{1e} &= s_s+\frac{1}{4}(F_{N-1}+F_N)T_{a,N}+\frac{1}{8}A_N T_{a,N}^2 \\
s_{2e} &= s_{1e}+(\frac{F_{N-1}+F_N}{2}+\frac{1}{2}A_N T_{a,N})T_{j,N}+\frac{1}{2}A_N T_{j,N}^2-\frac{1}{6}J_N T_{j,N}^3 \\
s_{3e} &= s_{2e}+F_N T_{f,N} \\
s_{4e} &= s_{3e}+F_N T_{j,N+1}+\frac{1}{6}J_{N+1}T_{j,N+1}^3 \\
s_{5e} &= s_{4e}+(F_N+\frac{1}{2}J_{N+1}T_{j,N+1}^2)T_{a,N+1}+\frac{1}{2}A_{N+1}T_{a,N}^2 \\
s_{6e} &= s_{5e}+(F_N+\frac{1}{2}J_{N+1}T_{j,N+1}^2+A_{N+1}T_{a,N+1})T_{j,N+1}+\frac{1}{2}A_{N+1}T_{j,N+1}^2-\frac{1}{6}J_{N+1}T_{j,N+1}^3
\end{aligned} \tag{B.31}$$

To obtain an explicit equation for s_{6e}, travel length equations are recursively substituted into the expression for s_{6e} which results in the following equation:

$$\begin{aligned}
s_{6e} =& \frac{1}{4}(F_{N-1}+F_N)T_{a,N}+\frac{1}{8}A_N T_{a,N}^2+(\frac{F_{N-1}+F_N}{2}+\frac{1}{2}A_N T_{a,N})T_{j,N}+\frac{1}{2}A_N T_{j,N}^2 \\
&-\frac{1}{6}J_N T_{j,N}^3+F_N T_{f,N}+F_N T_{j,N+1}+\frac{1}{6}J_{N+1}T_{j,N+1}^3+(F_N+\frac{1}{2}J_{N+1}T_{j,N+1}^2)T_{a,N+1} \\
&+\frac{1}{2}A_{N+1}T_{a,N}^2+(F_N+\frac{1}{2}J_{N+1}T_{j,N+1}^2+A_{N+1}T_{a,N+1})T_{j,N+1}+\frac{1}{2}A_{N+1}T_{j,N+1}^2 \\
&-\frac{1}{6}J_{N+1}T_{j,N+1}^3
\end{aligned} \tag{B.32}$$

Substituting in $J_k = A_k / T_{j,k}$ and grouping terms in Equation (B.32) results in the following equation:

$$\begin{aligned} s_{6e} &= F_{N-1}\left[\frac{1}{4}T_{a,N}+\frac{1}{2}T_{j,N}\right] \\ &+A_N\left[\frac{1}{8}T_{a,N}^2+\frac{1}{2}T_{a,N}T_{j,N}+\frac{1}{3}T_{j,N}^2\right] \\ &+F_N\left[\frac{1}{4}T_{a,N}+\frac{1}{2}T_{j,N}+T_{f,N}+2T_{j,N+1}+T_{a,N+1}\right] \\ &+A_{N+1}\left[\frac{1}{2}T_{a,N}^2+\frac{3}{2}T_{j,N+1}T_{a,N+1}+T_{j,N+1}^2\right] \end{aligned} \tag{B.33}$$

Substitute $T_{j,k}$ and $T_{a,k}$, Equations (B.1) and (B.9), respectively, into Equation (B.33).

$$\begin{aligned} s_{6e} &= F_{N-1}\left[\frac{1}{4}\left(\frac{F_N-F_{N-1}}{A_N}-\frac{A_N}{J_N}\right)+\frac{1}{2}\frac{A_N}{J_N}\right] \\ &+A_N\left[\frac{1}{8}\left(\frac{F_N-F_{N-1}}{A_N}-\frac{A_N}{J_N}\right)^2+\frac{1}{2}\left(\frac{F_N-F_{N-1}}{A_N}-\frac{A_N}{J_N}\right)\frac{A_N}{J_N}+\frac{1}{3}\frac{A_N^2}{J_N^2}\right] \\ &+F_N\left[\frac{1}{4}\left(\frac{F_N-F_{N-1}}{A_N}-\frac{A_N}{J_N}\right)+\frac{1}{2}\frac{A_N}{J_N}+T_{f,N}+2\frac{A_{N+1}}{J_{N+1}}+\left(\frac{F_{N+1}-F_N}{A_{N+1}}-\frac{A_{N+1}}{J_{N+1}}\right)\right] \\ &+A_{N+1}\left[\frac{1}{2}\left(\frac{F_{N+1}-F_N}{A_{N+1}}-\frac{A_{N+1}}{J_{N+1}}\right)^2+\frac{3}{2}\frac{A_{N+1}}{J_{N+1}}\left(\frac{F_{N+1}-F_N}{A_{N+1}}-\frac{A_{N+1}}{J_{N+1}}\right)+\frac{A_{N+1}^2}{J_{N+1}^2}\right] \end{aligned} \tag{B.34}$$

Expand and simplify Equation (B.34).

$$\begin{aligned} s_{6e} &= \frac{F_{N-1}(F_N-F_{N-1})}{4A_N}-\frac{F_{N-1}A_N}{4J_N}+\frac{F_{N-1}A_N}{2J_N} \\ &+\frac{(F_N-F_{N-1})}{8A_N}-\frac{2(F_N-F_{N-1})A_N}{8J_N}+\frac{A_N^3}{8J_N^2}+\frac{(F_N-F_{N-1})A_N}{2J_N}-\frac{A_N^3}{2J_N^2}+\frac{A_N^3}{3J_N^2} \\ &+\frac{F_N(F_N-F_{N-1})}{4A_N}-\frac{F_NA_N}{4J_N}+\frac{F_NA_N}{2J_N}+F_NT_{f,N}+\frac{2F_NA_{N+1}}{J_{N+1}}+\frac{F_N(F_{N+1}-F_N)}{A_{N+1}} \\ &-\frac{F_NA_{N+1}}{J_{N+1}}+\frac{(F_{N+1}-F_N)^2}{2A_{N+1}}-\frac{2(F_{N+1}-F_N)A_{N+1}}{2J_{N+1}}+\frac{A_{N+1}^3}{2J_{N+1}^2}+\frac{3(F_{N+1}-F_N)A_{N+1}}{2J_{N+1}} \\ &-\frac{3A_{N+1}^3}{2J_{N+1}^2}+\frac{A_{N+1}^3}{J_{N+1}^2} \end{aligned} \tag{B.35}$$

Thus,

$$\begin{aligned} s_{6e} = {} & \frac{F_{N-1}(F_N - F_{N-1})}{4A_N} + \frac{F_N A_N}{2J_N} + \frac{(F_N - F_{N-1})^2}{8A_N} - \frac{A_N^3}{24J_N^2} + \frac{F_N(F_N - F_{N-1})}{4A_N} \\ & + F_N T_{f,N} + \frac{F_N(F_{N+1} - F_N)}{A_{N+1}} + \frac{(F_{N+1} - F_N)^2}{2A_{N+1}} + \frac{F_N A_{N+1}}{J_{N+1}} + \frac{A_{N+1}(F_{N+1} - F_N)}{2J_{N+1}} \end{aligned} \tag{B.36}$$

Substituting the $\Delta_k = F_k - F_{k-1}$ into Equation (B.36) yields:

$$\begin{aligned} s_{6e} = {} & \frac{F_{N-1}\Delta_N}{4A_N} + \frac{\overbrace{F_N}^{*1}\Delta_N}{4A_N} + \frac{{\Delta_N}^2}{8A_N} + \frac{F_N A_N}{2J_N} - \frac{A_N^3}{24J_N^2} + F_N T_{f,N} \\ & + \frac{\overbrace{F_N}^{*2}\Delta_{N+1}}{A_{N+1}} + \frac{{\Delta_{N+1}}^2}{2A_{N+1}} + \frac{\overbrace{F_N}^{*2} A_{N+1}}{J_{N+1}} + \frac{A_{N+1}\Delta_{N+1}}{2J_{N+1}} \end{aligned} \tag{B.37}$$

Here we will substitute $F_N = F_{N-1} + \Delta_N$ into the first starred term (*1) and $F_N = F_{N+1} - \Delta_{N+1}$ into the second and third starred terms (*2) of Equation (B.37).

$$\begin{aligned} s_{6e} = {} & \frac{F_{N-1}\Delta_N}{4A_N} + \frac{\overbrace{(F_{N-1} + \Delta_N)}^{*1}\Delta_N}{4A_N} + \frac{{\Delta_N}^2}{8A_N} + \frac{F_N A_N}{2J_N} - \frac{A_N^3}{24J_N^2} + F_N T_{f,N} \\ & + \frac{\overbrace{(F_{N+1} - \Delta_{N+1})}^{*2}\Delta_{N+1}}{A_{N+1}} + \frac{{\Delta_{N+1}}^2}{2A_{N+1}} + \frac{\overbrace{(F_{N+1} - \Delta_{N+1})}^{*2}A_{N+1}}{J_{N+1}} + \frac{A_{N+1}\Delta_{N+1}}{2J_{N+1}} \\ & \qquad\qquad\qquad\qquad \downarrow \\ s_{6e} = {} & \frac{F_{N-1}\Delta_N}{2A_N} + \frac{3{\Delta_N}^2}{8A_N} + \frac{F_N A_N}{2J_N} - \frac{A_N^3}{24J_N^2} + F_N T_{f,N} \\ & + \frac{F_{N+1}\Delta_{N+1}}{A_{N+1}} - \frac{{\Delta_{N+1}}^2}{2A_{N+1}} + \frac{F_{N+1}A_{N+1}}{J_{N+1}} - \frac{A_{N+1}\Delta_{N+1}}{2J_{N+1}} \end{aligned} \tag{B.38}$$

Then setting the total distance traveled equal to the segment arc-length ($s_{6e} = L_N$) and solving for $T_{f,N}$ results in the following expression for the constant feedrate time duration for the *N*th segment:

$$T_{f,N} = \frac{L_N}{F_N} - \frac{1}{F_N}\left[\begin{array}{l} \dfrac{3\Delta_N{}^2}{8A_N} - \dfrac{A_N^3}{24J_N^2} + \dfrac{F_N A_N}{2J_N} + \dfrac{F_{N-1}\Delta_N}{2A_N} \\ + \dfrac{F_{N+1}\Delta_{N+1}}{A_{N+1}} - \dfrac{\Delta_{N+1}{}^2}{2A_{N+1}} + \dfrac{F_{N+1}A_{N+1}}{J_{N+1}} - \dfrac{A_{N+1}\Delta_{N+1}}{2J_{N+1}} \end{array}\right] \qquad \text{(B.39)}$$

Note that by definition of the final segment, the end feed F_{N+1} is equal to zero. It is still included here for completeness.

Kinematic Compatibility Conditions

Jerk Condition

The maximum jerk is limited by the sampling frequency of the CNC controller. Using Equation (B.1), the maximum allowable jerk magnitude is based on the specified maximum acceleration $(A_{\max})$ and sample period (T_s). The specified jerk (J_k) must be less than the maximum allowable jerk as follows:

$$J_k \le J_{\max} = \frac{A_{\max}}{T_s} \qquad \text{(B.40)}$$

Acceleration Condition

If the magnitude of acceleration for the constant acceleration phase is larger than what is required to change the feed from F_{k-1} to F_k, then the acceleration value is incompatible with the desired feed transition. For the acceleration value to be compatible, the time duration of the constant acceleration phase must be non-negative, that is $T_{a,k} \ge 0$. Using Equation (B.9), the condition on the acceleration magnitude (A_k) based on the specified jerk (J_k) and desired feedrates (F_{k-1}, F_k) is derived in the following manner:

$$
\begin{aligned}
0 &\leq T_{a,k} \\
0 &\leq \frac{F_k - F_{k-1}}{A_k} - \frac{A_k}{J_k} \\
\frac{A_k}{J_k} &\leq \frac{F_k - F_{k-1}}{A_k} \\
{A_k}^2 &\leq J_k (F_k - F_{k-1}) \\
|A_k| &\leq \sqrt{J_k (F_k - F_{k-1})}
\end{aligned}
\tag{B.41}
$$

Travel Length Condition (Feed Compatibility)

In order to have feed compatibility, the motion described by the specified values of acceleration (A_k, A_{k+1}), jerk (J_k, J_{k+1}) and feed (F_{k-1}, F_k, F_{k+1}) must be achievable within the available travel length. Therefore, after subtracting the distance required for the acceleration transients from the segment arc-length (L_k), the remaining travel length for the constant feedrate phase must be greater than or equal to zero, that is $(T_{f,k}F_k \geq 0)$. Using Equations (B.19), (B.29), and (B.39), the initial (*1*st), middle (*k*th), and final (*N*th) segments' feed compatibility equations are stated as follows:

$$
T_{f,1}F_1 = L_1 - \begin{bmatrix} \dfrac{F_0 A_1}{J_1} + \dfrac{F_0 \Delta_1}{A_1} + \dfrac{{\Delta_1}^2}{2A_1} + \dfrac{A_1 \Delta_1}{2J_1} \\ -\dfrac{3{\Delta_2}^2}{8A_2} + \dfrac{A_2^3}{24J_2^2} + \dfrac{F_1 A_2}{2J_2} + \dfrac{F_2 \Delta_2}{2A_2} \end{bmatrix} \geq 0
\tag{B.42}
$$

$$
T_{f,k}F_k = L_k - \begin{bmatrix} \dfrac{3{\Delta_k}^2}{8A_k} - \dfrac{A_k^3}{24J_k^2} + \dfrac{F_k A_k}{2J_k} + \dfrac{F_{k-1} \Delta_k}{2A_k} \\ -\dfrac{3{\Delta_{k+1}}^2}{8A_{k+1}} + \dfrac{A_{k+1}^3}{24J_{k+1}^2} + \dfrac{F_k A_{k+1}}{2J_{k+1}} + \dfrac{F_{k+1} \Delta_{k+1}}{2A_{k+1}} \end{bmatrix} \geq 0
\tag{B.43}
$$

$$
T_{f,N}F_N = L_N - \begin{bmatrix} \dfrac{3{\Delta_N}^2}{8A_N} - \dfrac{A_N^3}{24J_N^2} + \dfrac{F_N A_N}{2J_N} + \dfrac{F_{N-1} \Delta_N}{2A_N} \\ + \dfrac{F_{N+1} \Delta_{N+1}}{A_{N+1}} - \dfrac{{\Delta_{N+1}}^2}{2A_{N+1}} + \dfrac{F_{N+1} A_{N+1}}{J_{N+1}} - \dfrac{A_{N+1} \Delta_{N+1}}{2J_{N+1}} \end{bmatrix} \geq 0
\tag{B.44}
$$

This section has derived expressions for the time durations of each type of phase (constant non-zero jerk, constant non-zero acceleration, and constant non-zero feedrate) and expressions for the kinematic compatibility conditions (jerk, acceleration, and travel length). These equations are used in implementing a continuous feedrate modulation strategy for an *N*-segment toolpath which was discussed in Chapter 4.