FIVE-AXIS TRAJECTORY GENERATION METHODS
By
Burak Sencer
B. Sc., Istanbul Technical University, 2003
A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF APPLIED SCIENCE
in
THE FACULTY OF GRADUATE STUDIES(MECHANICAL ENGINEERING)


#### Abstract

Five-axis machining has been widely adopted manufacturing parts with sculptured surfaces in the aerospace, die and mold and other industries. Five-axis machine tools have a kinematic structure compromising three Cartesian axes couples with two rotary joints enabling the tool to reach any given position and the orientation on the workpiece surface. Five-axis motion allows the orientation of the tool along the toolpaths with varying curvatures and sculptured surfaces. However, optimal trajectory generation methods are required in achieving a smooth coordinated motion of the tool tip and tool axis orientation, which is essential in reducing the fluctuations in the feed, accelerate and jerk of the drives.

In this thesis, a toolpath generation algorithm has been developed for smooth trajectories based on compressing the $\mathrm{C}^{0}$ continuous Numerical Control (NC) motion commands into continuous $b$-spline format. With the aid of the $b$-spline toolpath representation, the discontinuities and unwanted acceleration harmonic in the multi-axis motion are reduces. The tool position and orientation vectors are interpolated continuously enabling a smooth and synchronized motion of the drives. In order to achieve higher smoothness in the change of feed direction, the spline toolpath is optimized by solving the linear quadratic minimization problem during the least squares fit. The resultant angular motion of the tool is further smoothed by minimizing the integral square of third derivative of the orientation spline using a non-linear optimization technique.

The Feed Correction Polynomial algorithm is implemented in the real time interpolation of the generated $b$-spline toolpath yielding a consistent feedrate profile during contouring operations. It is shown that feed correction method exhibits reduced feedrate fluctuations when compared to the widely used Taylor Series approximations.

A jerk continuous feedrate profile in terms of $\mathrm{C}^{3}$ quintic spline is adapted for five-axis trajectory generation. The feedrate profile is generated utilizing the minimum jerk criteria for the smoothness of the motion and tracking accuracy. The feed profile is then optimized to achieve minimum cycle time while adhering to the machine tool's velocity, acceleration and jerk limitations.

The developed trajectory generation algorithms have been simulated using the kinematics of a widely used five-axis machine tool promising potential applications in rough machining.


## Table of Contents

Abstract ..... ii
Table of Contents ..... iii
List of Tables ..... vi
List of Figures ..... vii
Nomenclature ..... ix
Acknowledgment ..... xiv
Chapter 1 Introduction ..... 1
Chapter 2 Literature Review ..... 3
2.1. Overview ..... 3
2.2. Spline Properties ..... 5
2.3. Spline Fitting for Sculptured Machining ..... 8
2.4. Trajectory Generation for 3 Axis Motions ..... 12
2.5. Critical Issues on 5-Axis Spline Trajectory Generation ..... 13
Chapter 3 Spline Toolpath Generation from Discrete Toolpath Data for 5-Axis
Machining ..... 18
3.1. Introduction ..... 18
3.2. B-Spline Background ..... 20
3.3. Double B-Spline Polynomial Representation ..... 24
3.4. Generation of Lower B-Spline ..... 27
3.4.1. Tolerance Definition and Parameter Correction ..... 33
3.5. Generation of the Upper Spline ..... 39
3.6. Smoothing the Double Spline Toolpath ..... 42
3.6.1. Introduction ..... 42
3.6.2. Smoothing the Lower B-Spline ..... 43
3.6.3. Angular Jerk Minimization ..... 47
3.7. Simulation Results ..... 50
3.8. Summary ..... 63
Chapter 4 Interpolation and Feedrate Profiling for 5-Axis Machining ..... 64
4.1. Introduction ..... 64
4.2. Interpolation with Minimal Feed Fluctuation ..... 64
4.2.1. Taylor Series Approximation ..... 66
4.2.2. Feed Correction Polynomial Method ..... 70
4.2.3. Iterative Solution of the Spline Parameter ..... 77
4.3. Feedrate Profiling for 5-Axis Machine Tools ..... 79
4.3.1. Minimum Jerk Feed Profile ..... 81
4.3.2. Velocity, Acceleration and Jerk Constraints ..... 84
4.3.3. Inverse Kinematics of a Rotary Tilting Table 5-Axis Machine Tool ..... 84
4.3.3.1 Rotational Motions ..... 88
4.3.3.2 Cartesian Motions ..... 91
4.3.4. Formulation of the Constraints ..... 95
4.3.5. Solution of the Optimization Problem ..... 97
4.4. Simulation Results ..... 99
4.4.1. Comparison of the Interpolation Methods ..... 99
4.4.2. Comparison of Constant and Optimized Feedrate Profiles ..... 102
4.4.3. Summary ..... 105
Bibliography ..... 111
Appendix A Formulation of the Minimum Jerk Feed Profile ..... 120
Appendix B Formulation of Gradients of Feedrate Optimization ..... 126
B.1. Gradient of the Objective ..... 125
B.2. Gradient of the Feed Profile Coefficients ..... 126
B.3. Gradient of the Kinematic Profiles ..... 127
B.4. Gradients of the Constraints ..... 129

## List of Tables

Table 3.1: Comparison of Curvatures ..... 56
Table 4.1: Rotary Axis Angles ..... 91
Table 4.2: Summary of Interpolation Results ..... 100
Table 4.3: Kinematic Limits of 5-Axis Machine Tool ..... 103

## List of Figures

Figure 2.1: Reference Trajectory Generation ..... 3
Figure 2.2: Quintic Polynomial Spline ..... 6
Figure 2.3: Bezier Curve ..... 7
Figure 2.4: B-spline curve ..... 8
Figure 2.5: Tool Tip and Tool Orientation Vectors ..... 14
Figure 2.6: a) Position Spline --- b) Orientation Spline ..... 15
Figure 3.1: Flow Chart of Optimum B-spline Toolpath Generation ..... 19
Figure 3.2: B-spline Toolpath ..... 21
Figure 3.3: B-spline Basis Functions ..... 23
Figure 3.4: Double B-Spline Toolpath Representation ..... 25
Figure 3.5: Upper and Lower B-splines ..... 26
Figure 3.6: Initial Spline Parameter Calculation ..... 28
Figure 3.7: Tolerance Definition for Lower B-spline Generation ..... 34
Figure 3.8: Spline Parameter Correction ..... 35
Figure 3.9: Zero Tolerance Definitions a) Zero Cosine --- b) Point Coincidence ..... 36
Figure 3.10: Lower B-spline Fitting Algorithm ..... 38
Figure 3.12: Double B-spline Generation Algorithm ..... 41
Figure 3.13: Tool Axis Vector ..... 47
Figure 3.14: Test Double B-spline Toolpaths Generated from Discrete Points ..... 52
Figure 3.15: Fitting Errors of the B-Spline Toolpaths ..... 53
Figure 3.16: Multi Objective Least Squares Smoothening of a B-spline - a) Regular Least Squares Fit, b-d) Optimized Least Squares Fit Comparisons ..... 55
Table 3.1: Comparison of Curvatures ..... 56
Figure 3.17: $2^{\text {nd }}$ and $3{ }^{\text {rd }}$ Derivative Comparison of Smoothened B-splines ..... 56
Figure 3.18: Comparison of Optimized Lower B-Spline ..... 58
Figure 3.19: Angular Kinematic Profiles ..... 60
Figure 3.20: Rotary Axis Kinematic Profiles ..... 61
Figure 3.21: Linear Axis Kinematic Profiles ..... 62
Figure 4.1: Real-time Spline Interpolation ..... 65
Figure 4.2: Basis Function Computation a) Recursive Calculation of Basis Functions --- b)
Recursive Calculation of Basis Function Derivatives ..... 69
Figure 4.3: B-spline Knot Spans and Segmentation ..... 71
Figure 4.4: Feed Correction Polynomial for B-spline Toolpath ..... 73
Figure 4.5: Real Time Interpolation Methods ..... 77
Figure 4.6: Overview of Feedrate Profiling ..... 80
Figure 4.7: Kinematic Profiles on the B-spline Toolpath ..... 82
Figure 4.8: Kinematic Chain of the Rotary Tilting Table 5-Axis Machine Tool ..... 85
Figure 4.9: a) Translation Motion --- b) Rotational Motion ..... 86
Figure 4.10: a) Table Orientation Before Cutting --- b) Table Orientation During Cutting ..... 89
Table 4.1 Rotary Axis Joint Angles ..... 91
Figure 4.11: Kinematic Transformation ..... 94
Figure 4.12: Constraint Evaluation ..... 98
Figure 4.13: B-spline Toolpath ..... 101
Figure 4.14: Feedrate Fluctuations ..... 101
Figure 4.15: B-spline Toolpath ..... 103
Figure 4.16: Optimized and Unoptimized Feedrate Profiles ..... 104
Figure 4.17: Optimized and Unoptimized Cartesian Axis Profiles ..... 104
Figure 4.18: Optimized and Unoptimized Rotary Axis Profiles ..... 105

## Nomenclature

| $A=A(u)$ | $=$ tool axis spline |
| :---: | :---: |
| $A_{k}^{f}, B_{k}^{f}, \ldots H_{k}^{f}$ | $=$ feed correction polynomial coeff. |
| $a_{k}^{f}, b_{k}^{f}, \ldots, h_{k}^{f}$ | $=$ normalized feed correction polynomial coeff. |
| $A_{k}, B_{k}, \ldots, C_{k}$ | $=$ quintic spline displacement profile coeff. |
| $a_{k}, b_{k}, \ldots, c_{k}$ | $=$ quintic spline displacement profile normalized coeff. |
| $A_{x}, A_{y}, A_{z}$ | $=\mathrm{x}, \mathrm{y}, \mathrm{z}$ axis components of tool axis spline |
| $a_{X \text { max }}, a_{Y_{\text {max }}}, a_{Z_{\text {max }}}, a_{\theta_{A} \max }, a_{\theta_{C} \max }=\mathrm{x}, \mathrm{y}, \mathrm{z}, \mathrm{A}, \mathrm{C}$ axis acceleration limits |  |
| C | $=$ vector of inequality constraints |
| $\bar{D}$ | $=$ vector between estimated and actual data points |
| $e$ | = error, position and parameter |
| $e_{\text {max }}$ | $=$ maximum tool tip positioning error |
| $e_{\theta}$ | = orientation error |
| $e_{\text {өmax }}$ | $=$ maximum tool orientation error |
| $f_{\text {max }}$ | $=$ maximum tangential feedrate |
| H | $=$ relative distance between lower and upper B-splines |
| $H_{o r}^{R e f}$ | $=$ homogeneous transformation matrices between tool and |
| reference frames |  |
| $H_{W P}^{\text {Ref }}$ | =homogeneous transformation matrices between workpiece |
| and reference frames |  |
| $J_{K}$ | $=$ objective function for smoothing |
| $J_{L}$ | $=$ objective function of least squares fit |
| $\bar{J}_{L}$ | $=$ augmented objective function for least squares |
| $J_{V}$ | $=$ objective function for minimum jerk |
| $J_{K-x, y, z}$ | = axis components of smoothing objective function |
| $J_{L_{x, y, z}}$ | $=$ axis components of the least squares objective function |


| $J_{o_{\text {utu }}}$ | $=$ objective function for minimum angular jerk |
| :---: | :---: |
| $J_{V}$ | $=$ objective function for minimum jerk disp. profile |
| $\bar{J}_{V}$ | $=$ augmented objective function for minimum jerk disp profile |
| $J_{V_{k}}$ | $=$ segment objective function for minimum Jerk disp profile |
| $j_{X \text { max }}, j_{Y \text { max }}, j_{Z_{\text {max }}}, j_{\theta_{A} \text { max }}, j_{\theta_{C} \text { max }}=\mathrm{x}, \mathrm{y}, \mathrm{z}, \mathrm{A}, \mathrm{C}$ axes jerk limits |  |
| K | $=$ weight matrix |
| $L$ | = constraint matrix |
| M | $=$ number of fitting data points -1 |
| $m$ | $=$ number of knots -1 |
| $\dot{M}(t), \ddot{M}(t), \dddot{M}(t)$ | $=\mathrm{x}, \mathrm{y}, \mathrm{z}, \mathrm{A}, \mathrm{C}$ axes velocity, acceleration and jerk profiles |
| $M_{s,} M_{s s}, M_{s s s}$ | $=$ geometric derivatives of the toolpath |
| $n$ | $=$ number of control points -1 |
| $N_{i, 0}, N_{i, 1}, N_{i, 2}, \ldots, N_{i, p}$ | $=$ basis functions |
| Ns | $=$ number of segments (internal knot spans) |
| $N_{u}, N_{u u}, N_{u u u}$ | $=1^{\text {st }}, 2^{\text {nd }}, 3^{\text {rd }}$ derivatives of the basis functions |
| $O=O(u)$ | = Orientation spline |
| $O^{r}$ | $=$ reference orientation |
| $O_{x}^{r}, O_{y}^{r}, O_{z}^{r}$ | $=\mathrm{x}, \mathrm{y}, \mathrm{z}$ axis components of reference orientation |
| $O^{\text {tool }}$ | = fixed tool orientation |
| $O T_{x}, O T_{y}, O t_{z}$ | $=$ tool frame axial offset components |
| $O_{u}, O_{u u}, O_{\text {uuu }}$ | $=1^{\text {st }}, 2^{\text {nd }}$ and $3^{\text {rd }}$ derivative of the orientation Spline |
| $O_{x}, O_{y}, O_{z}$ | $=\mathrm{x}, \mathrm{y}, \mathrm{z}$ axis components of orientation spline |
| $P_{i}$ | $=$ control points of lower(tool-tip) B-spline |
| $p$ | $=$ degree of the B -spline |
| $P(u)$ | $=$ Tool tip (Lower) B-spline |
| $P_{u}, P_{u u}, P_{u u u}$ | $=1^{\text {st }}, 2^{\text {nd }}$ and $3^{\text {rd }}$ derivative of Lower B-spline |
| $P_{x}, P_{y}, P_{z}$ | $=$ control points of lower B -spline $\mathrm{x}, \mathrm{y}, \mathrm{z}$ axis components |


| $Q_{i}$ | $=$ control points of upper B-spline |
| :---: | :---: |
| $Q(u)$ | $=$ upper B-spline |
| $q_{0}^{t}, q_{1}^{t}, \ldots, q_{k}^{t}, q_{M}^{t}$ | $=$ fitting data points |
| $\hat{q}_{x_{k}}^{t}$ | $=\mathrm{x}$ axis component of estimated data points |
| $q_{x_{k}}^{t}, q_{y_{k}}^{t}, q_{z_{k}}^{t}$ | $=\mathrm{x}, \mathrm{y}, \mathrm{z}$ axis components of actual Data Points |
| $Q_{u}, Q_{u u}, Q_{\text {uuu }}$ | = first, second and third derivative of Upper B-spline |
| $Q_{x}, Q_{y}, Q_{z}$ | $=$ control points of upper B -spline $\mathrm{x}, \mathrm{y}, \mathrm{z}$ axis components |
| $r$ | $=$ total number of constraints |
| $\operatorname{Re} f_{x}, \operatorname{Re} f_{y}, \operatorname{Re} f_{z}$ | $=$ reference frame axial offset components |
| $R_{x}, R_{y}, R_{z}$ | $=$ rotary ( C ) frame axial offset components |
| $s=s(t)$ | $=$ displacement profile |
| $\dot{s}_{k}(\nu)$ | $=$ velocity profile |
| $\ddot{s}_{k}(v)$ | $=$ tangential acceleration profile |
| $\dddot{s}_{k}(\nu)$ | $=$ jerk profile |
| $\dot{s}_{\text {final }}, \ddot{s}_{\text {final }}, \dddot{\dddot{s}}_{\text {final }}$ | $=$ final velocity, acceleration and jerk conditions |
| $\dot{s}_{\text {init }}, \ddot{s}_{\text {init }}, \dddot{s}_{\text {init }}$ | $=$ initial velocity, acceleration and jerk conditions |
| $S_{k}$ | $=$ segment arc travel lengths |
| $S p_{x}, S p_{y,} S p_{z}$ | $=$ spindle frame axial offsets |
| $S_{\text {total }}$ | $=$ total arc length spline |
| $S_{x,} S_{y}, S_{z}$ | $=$ table frame axial offsets |
| $T_{0}, \ldots, T_{k}, \ldots, T_{N_{s}-1}$ | $=$ segment time durations |
| $T_{\text {max }}$ | $=$ maximum expected segment time duration |
| Ts | = servo control loop sampling interval |
| $T_{\sigma}$ | $=$ total travel time |
| $T_{x}, T_{y}, T_{z}$ | $=$ tilting (A) frame axial offsets components |
| $U$ | $=$ knot vector of the B-spline |

u
$\bar{u}$
$u_{\text {test }}$
$V=V(t)$
$v_{X \text { max }}, v_{Y_{\text {max }}}, v_{Z \text { max }}, v_{\theta_{A} \max }, v_{\theta_{C} \text { max }}=$ axis velocity limits
$W p_{x} W p_{y}, W p_{z}$
$X, Y, Z$
$x^{h}, y^{h}, z^{h}$
$u_{\text {prev }}$
$x^{t}, y^{t}, z^{t}$
$x_{\text {test }}, y_{\text {test }}, z_{\text {test }}$
$=$ spline parameter
= estimated B -spline parameter
$=$ previous spline parameter
$=$ tested spline parameter
$=$ tangential feedrate
$=$ workpiece frame axial offsets
$=$ previous position commands
$=$ tested position commands
$=$ parameter values corresponding to data points
$=\mathrm{x}, \mathrm{y}, \mathrm{z}$ axis position commands or indices
$=\mathrm{x}, \mathrm{y}, \mathrm{z}$ axis components of Upper B-spline
$=\mathrm{x}, \mathrm{y}, \mathrm{z}$ axis components of Lower(tool-tip) B-spline

| $\Phi$ | = regressor matrix |
| :---: | :---: |
| $\xi$ | $=$ constraint vector for quadratic programming |
| $\Delta$ | $=$ vector of Lagrange multipliers |
| $\varepsilon$ | $=$ relaxation tolerance for nonlinear optimization |
| $\varepsilon_{1}$ | $=$ measure of the Euclidean distance |
| $\varepsilon_{2}$ | = zero cosine value |
| $\theta_{h}$ | $=$ vector of upper B-spline optimization variables |
| $\alpha$ | $=$ fitting effect coefficient |
| $\gamma$ | $=$ smoothing effect coefficient |
| $\varepsilon$ | $=$ relaxation tolerance |
| $\sigma$ | = normalized arc length travel of segment |
| $\theta_{f}$ | $=$ vector of Feed Correction Poly. optimization variables |
| $\lambda_{k}$ | $=$ normalized time duration |
| $\theta_{f}$ | $=$ vector of feed correction polynomial coefficients |
| $\theta_{k}$ | $=$ minimum jerk disp. profile optimization variables |
| $\theta_{h}$ | $=$ minimum jerk orientaiton profile optimization variables |
| $\tau$ | $=$ spline displacement profile time parameter |
| $v$ | $=$ normalized time parameter |
| $\theta_{A}$ | $=$ tilting axis (A) rotation angle |
| $\theta_{C}$ | $=$ rotary axis (C) rotation angle |
| $\Delta s$ | $=$ arc increment |
| $\Delta u$ | = parameter increment |
| $\mu$ | = knot |
| $\hat{\mu}$ | $=$ inserted knot |

## Acknowledgment

I would like to express my sincere appreciation to my research supervisor Dr. Yusuf Altintas for his valuable instruction, guidance, and support which he has provided throughout my graduate studies at University of British Columbia. I would like to thank Nesrin Altintas for her hospitality.

I wish to thank my colleagues in the Manufacturing Automation Laboratory for sharing their knowledge and experience with me. I also feel very luckily to gain such an exceptional friend, Doruk. He was with me especially during the tough times and made my life colorful and enjoyable.

Finally, I am deeply grateful to my beloved family, my mother Serap and my father Necip for their love, encouragement unweaving support and patience they have provided me throughout my entire life and abroad studies. This thesis and all of my previous success are dedicated to them.

## Chapter 1

## Introduction

The Computer Aided Design (CAD) technology has led to smooth spline representation of space curves and complex free-form surfaces with steep curvatures, commonly used in designing dies, mold and aerospace parts with sculptures surfaces. Bezier splines (b-splines) and non-uniform rational b-splines (NURBS) are extensively used because of their dexterity in modifying and representing curved geometries with high smoothness. Sculptured machining of free-form surfaces by five-axis computer-numerically-controlled (CNC) machine tools has received significant attention in recent years since it provides optimal tool posture and orientation, not possible with three-axis Cartesian machines. In particular, surfaces with varying curvatures are machined on five-axis machining centers where the tool orientation is adjusted with varying curvatures are machined on five-axis machining centers where the tool orientation is adjusted continuously resulting in a smoother surface finish. Furthermore, on such machines, machining of deep cavities and inclined drilling are performed using shorter cutters, leading to less vibrations and higher material removal rates.

However, current practice in five-Axis machining approximates the contours with straight line segments yielding large $\mathrm{C}^{0}$ continuous NC toolpaths, composed of tool position and orientation vectors. The linear segmentation technique leads to a rough surface finish and forces the machine tool to decelerate, accelerate or stop at each segment junction. In particular, for machines with rotary axes located on the table, the axes have to move and support significant masses during machining. As a result, the acceleration limits of such fiveaxis machines are reduces, increasing the machining cycle times.

Tool motion with abrupt deceleration and accelerations cause contouring errors found at locations where the transients such as tangency or curvature discontinuities occur in the reference trajectory. Such discontinuous feed motions may excite the natural modes of the mechanical structure of the mechanical structure and servo control system, characterized by vibration marks on the part surface and degradation of the axis tracking performance. The discontinuous toolpath must, therefore, be replaced with spline representations for smooth trajectory generation spline toolpaths facilitate the delivery of the desired, smooth, high
speed feed motion by enforcing acceleration and jerk continuity in the trajectory. In this thesis, the smooth and continuous toolpath requirements are implemented using $b$-splines that are capable of interpolating tool position and orientation vectors, and generated by compressing the original NC codes.

Unfortunately, spline trajectories exhibit undesirable feedrate fluctuations during real time interpolation. When the curves are interpolated at constant parameter increment, fluctuations in the tangential feedrate occur, which reduce the smoothness of the coordinated multi-axis motion. The appropriate spline parameter must be calculated in order to realize the desired feed profile during machining operation. b-spline toolpaths are used with the Feed Correction Polynomial method in order to smoothly parameterize the curved paths for fast and accurate real time interpolation.

The feedrate profile of the trajectory must also maximize smoothness for tracking performance and minimize cycle time for productivity. The optimized smoothness can be achieved by minimizing the jerk discontinuity of the overall trajectory. However, due to the kinematic limits of the drives and tracking performance of the servo system, the tangential feedrate profile is generated conservatively, compromising the minimum cycle time objective. The inverse kinematics transformation of the five-axis machine tool structure further diminishes the linear relationship between the tool motion in workpiece coordinates and axis kinematic profiles. As a result, the optimum scheduling of the feedrate can only be achieved by considering the curvature of the toolpath, the change in the orientation and the kinematic transformation, while adhering the machine tool's velocity, acceleration and jerk limitations. By adapting the feedrate profiling to accommodate these non-linear factors, the high speed tracking and contouring accuracy during the execution of the optimal feed motion can be preserved.

The thesis is organized as follows: The review of the related literature is presented in Chapter 2, followed by smooth spline toolpath generation method in Chapter 3. The real time interpolation of the b-spline toolpath and the feedrate scheduling algorithm for five-axis Machining is given in Chapter 4. The conclusions and possible future research directions are discussed in Chapter 5.

## Chapter 2

## Literature Review

### 2.1. Overview

Reference trajectory generation plays a significant role in five-axis computer numerical controlled (CNC) machine tools to acquire the desired part quality. Generation of the reference motion commands to the machine tool's axis drives consist of three main stages: Planning the machining toolpath on the part, imposing the tangential feedrate profile and real time interpolation of the position commands within the servo loop control frequency.


Figure 2.1: Reference Trajectory Generation

The attempt in generating a smooth trajectory starts at the toolpath planning step. Small discontinuities in reference toolpaths can result in undesirable high frequency harmonics in reference trajectory that excite the natural modes of the mechanical structure or servo control system. The resultant high frequency components can cause actuator saturations and axis tracking errors causing violations of the machining tolerances on the manufactured parts. Furthermore, employing only linear and circular interpolation techniques to machine complex shapes such as dies, molds, turbine blades, and aerospace parts has serious limitations in productivity since the machine tool must decelerate/accelerate or stop between consecutive G codes [42]. Some amount of work has been done to overcome these problems by developing spline tool path definitions for $3[78,79,80]$ and five-axis machine tools [1, $29,55]$. Spline toolpaths are parametric curves where the spline parameter does not have an explicit relationship to the total travel arc length. Therefore, ensuring the desired feedrate profile on a spline tool path requires precise calculation of the spline parameter increment during real time interpolation. Feed fluctuations posed by improper calculation of the curve parameter increase the contouring errors. A significant amount of research has been done in real time spline interpolation techniques [20, 42,50]. On the other hand, the kinematic profiles of individual axes are defined by the feedrate profile of the trajectory. Thus, the feedrate profile must avoid any discontinuity in velocity, acceleration and jerk. To enhance the tracking performance of the servo systems, overall jerk content of the trajectory needs to be minimized and to increase the productivity the feed profile must be scheduled for minimum cycle time [3] while respecting the machine tool's kinematic limits. This is a challenge particularly required to be tacked in five-axis high speed machining centers where the kinematic transformations are significant. The literature reviewed about the trajectory generation techniques for multi axis machining is presented in the following sections.

This chapter organized as follows: the spline properties used in tool path definition are introduced in Section 2.2. This is followed by a literature survey of the spline fitting techniques for sculptured surface machining. The research on trajectory generation for 3Axis machining is briefly presented in Section 2.3. A literature review on 5-Axis spline interpolation and motion generation is covered in Section 2.4.

### 2.2. Spline Properties

Splines are powerful tools that have intensively being used to design smooth and complex geometries such as dies, molds, gas tribune blades and aeronautical parts in the computer aided design (CAD) stage. The most conventional method of presenting spline curves is using implicit equations [27] and parametric functions [58]. However, the majority of CAD systems use the parametric forms of curves instead of implicit since it is practical for part design and control of multi axis machine tools where each axis can be driven individually. Independent from their parametric representation, splines in computer aided design and manufacturing (CAM) systems can be grouped in three main categories:

- Polynomial splines,
- Bezier Curves and
- B-splines.

Cubic and Quintic splines are extensively used curve formats included in the polynomial spline category. Their definition bases on exactly interpolating certain design points (control knots) with piecewise polynomials. The polynomial curve segments are constructed in a way that the position and derivative continuity throughout the composite curve is preserved. A part from the cubic polynomial splines, quintic spline has taken emphasis in the area of CNC tool path generation from the researchers because of their advantageous parameterization [78] for real time interpolation and third order ( $\mathrm{C}^{3}$ ) continuity allowing implementation of jerk continuous trajectories [80]. The quintic spline represented in Fig. 2.2 is composed of $N$ - $I$ piecewise curve segments $S_{1}, \ldots, S_{N}$ interpolating the $N$ reference points $P_{1}, \ldots, P_{N-1}$. It is well defined in the polynomial form of

$$
P_{i}(u)=A_{i} u^{5}+B_{i} u^{4}+C_{i} u^{3}+D_{i} u^{2}+E_{i} u+F_{i}
$$

where for two axis cartesian case, $S_{i}=\left[\begin{array}{ll}S_{x i} & S_{y i}\end{array}\right]^{T}$ and $A_{i}=\left[\begin{array}{ll}A_{x i} & A_{y i}\end{array}\right]^{T}, \ldots, F_{i}=\left[\begin{array}{ll}F_{x i} & F_{y i}\end{array}\right]^{T}$. The unknown coefficients of quintic splines $\left(A_{i}, B_{i}, \ldots, F_{i}\right)$ are found by imposing position $\left(\mathrm{C}^{0}\right)$, and higher order continuity boundary conditions between consecutive piecewise curve
segments. However, the affect of coefficients on the spline shape is totally unpredictable, which makes the polynomial splines unsuitable for geometric design.


Figure 2.2: Quintic Polynomial Spline

Bezier curves are being extensively exploited in CAD because it merges the comprehensiveness of the parametrical format with the design functionality. Their definition is clearly presented by Piegl [26] in terms of basis (blending) functions $B_{i, p}(u)$ and control points $P_{i}=\left[\begin{array}{ll}P_{x i} & P_{y i}\end{array}\right]^{T}$ as,

$$
P(u)=\sum_{i=0}^{p} B_{i, p}(u) P_{i}
$$

where $p$ is the degree of the curve, which also specifies the number of total control points included in the curve definition. The blending functions are given in Bernstein form [26] by:

$$
B_{i, p}(u)=\frac{p!}{i!(p-i)!} u^{i}(1-p)^{p-i} \quad u \in\left[\begin{array}{ll}
0 & 1
\end{array}\right]
$$

As shown in Fig. 2.3, the Bezier curve interpolates only the first and last control points. Its complete shape is determined by the positions of the remaining points. Therefore, the shape can easily be adjusted by modification of any control point making it practical for geometric design.


Figure 2.3: Bezier Curve
On the other hand, large number of control points is required to generate complex surfaces resulting higher order Bezier curves. This drawback is defeated with b-splines introduced by de Boor [58]. Similar to Bezier curves, b-splines (See Fig. 2.4) are also defined by control points and basis functions. However, the number of control points and the degree can be selected independently allowing designers to work with lower degree curves. Furthermore, the complete spline is segmented into parameter intervals by a knot vector ( $U$ ), which enables local modification of the curve by using the nearest control points (local modification property). A $\mathrm{p}^{\text {th }}$ order b -spline curve is defined by,

$$
P(u)=\sum_{i=0}^{n} N_{i, n}(u) P_{i}
$$

where $N_{i, n}(u)$ are the new basis functions evaluated recursively using the de Boor algorithm [58].


Figure 2.4: B-spline curve

### 2.3. Spline Fitting for Sculptured Machining

It was a milestone for sculptured machining to represent complex parts with smooth curves, namely quintic, Bezier and the $b$-splines. But sending this geometric information with minimal lost to the machine tool in smooth recognizable formats such as splines is vital for high speed machining, since higher and smoother feedrate profiles as well as better tracking performance can be obtained.

However, the common practice in generating toolpaths for machining sculptured surfaces on 3 and five-axis machining centers bases on approximating the original surface with lines and sending this cutter location data stacked in a G code to the CNC Executive [62, 63]. Previous researchers Koren and Lo [42], and Yang and Kong [86] have demonstrated the following limitations of this linear interpolation method: Velocity discontinuities at the linear segment junctions lead to higher accelerations, poor surface finish, lower part accuracy and longer machining times.

Jouaneh et. al. [35] suggested using arc segments to connect the linear segments and to smooth the corner of the tool path in order to obtain the velocity continuity $\left(\mathrm{C}^{1}\right)$ at the
junctions making it easier to track. Similarly, Hua et. al. [61] used only arcs to discretize the curved contours for 3-Axis high speed machining. Yeung [91] pointed out that quintic splines can be used to connect linear segments, which allows acceleration continuity at the transitions and showed that the tracking performance of the servo system is improved tremendously.

Instead of linear and arc interpolation techniques, spline refitting and direct spline trajectory generation methods have been widely adapted in machining sculptured surfaces for HSM. Especially, cubic and quintic splines trajectories are utilized for real time toolpath interpolation because of their advantages in real time interpolation. Wang and Yang [78] introduced $\mathrm{C}^{2}$ cubic and quintic splines enabling acceleration continuous feed profiles. They pointed out that when splines are interpolated with constant chord length increment, feedrate fluctuations occur due to the discrepancy between the chord length parameter and the traveled arc length increment. They brought up the idea of fitting quintic splines with NearArc Length parameterization resulting in smaller feedrate fluctuations and smoother geometric derivatives. Later, Wang et. al. [79, 80] improved this methodology by preserving the continuity of third derivatives at the consecutive spline segment junctions. This improved technique is called Approximately Arc-Length Parameterization with $\mathrm{C}^{3}$ and reduced the feedrate fluctuations yielding smoother geometric derivatives. Erkorkmaz and Altintas [23] proposed Optimally Arc Length Parameterized quintic splines. This approach is based on optimizing the spline parameters in such a way that the discrepancy between the spline chord length parameter and traveled arc length increment is minimized, resulting less feedrate fluctuations compared to Near Arc Length parameterization.

Furthermore, research has been done in interpolating the quintic spline toolpath in real time when the parameterization is arbitrary. Shipitalni et. al. [67] proposed an iterative interpolation technique using the first order Taylor series approximation in order to determine the spline parameter at constant feedrate. However, this approach was not capable of eliminating the feed fluctuations in case of sharp curvature changes. Later, Lin [50] used second order approximation reporting two order of magnitude reduction in the feed fluctuations compared to the first order. Altintas and Erkorkmaz [21] developed an iterative method requiring the numerical solution of the root of a high order polynomial relating the arc increment to the parameter increment. They also used an additional polynomial to express
the nonlinearity between the spline parameter and the arc length, which is called Feed Correction Polynomial [19] . By the aid of this polynomial, the spline parameter can be computed efficiently in real time acquiring minimum feed fluctuation.

Besides the quintic polynomial splines, b-splines are widely used in generating parametric surfaces from discrete measurement data in reverse engineering [43, 65, 66], and b-spline trajectories for machine tools [5, 33, 47, 48] and robots [24, 92]. Since b-splines are capable of representing geometries with higher smoothness and continuity [73] compared to quintic polynomials, they are more suitable for high speed machining purposes. Lai et. al. [46] utilized $b$-splines to generate sculpture surfaces from laser-scan measurement data and extracted toolpaths for 3-Axis machining. He used a non-linear least squares technique to fit the axis splines within acceptable fitting tolerances. Later, Elkeran and El-Baz [17] used the idea to directly generate non uniform rational b-splines (NURBS) linear G code blocks using least squares. They fed the NC unit with NURBS and showed that consistent feed rates can be achieved as compared to the linear interpolation. Yau and Kuo [89] used the same method to compress $G$ codes into $b$-spline formats, which can be accepted by different machine tool NC units such as Fanuc and Siemens. Even those sophisticated CNC executives are equipped nowadays with build - in functions that compress numerous blocks of $G$ codes into smooth axis b-splines for HSM [70].

However, the quality of the curves as well as their real time interpolation technique is very important for the tracking performance and surface quality. These problems are addressed from two directions: generation of smooth axis splines and accurate interpolation of the b-splines in real time. Generally speaking, smooth curves with less change in the curvature and jerk content are desired since they are easier to track by the servo system especially when traveled at high feedrates. The problem of curve quality is first addressed by Horn [32]. He showed that maximum smoothness of the curve is achieved when the square integral of the curvature is minimized. Hohenberger and Reudig [31] used the curvature cost function and proposed an unconstrained iterative optimization scheme to approximate measurement data with smooth b-spline surfaces. They pointed out that smoother curves can be obtained while sacrificing from the fitting accuracy. Bercover [ 2,30 ] used various energy functionals for curve smoothness based on optimizing the $b$-spline control points as well as the knot vectors. On the other hand, Vassilev [75] and others [26,56] studied the b-spline
fitting while pursuing less computationally expensive methods for computer graphics. They investigated the affect of cost functions formulated by square sum of derivatives with respect to the curve parameter for smoothness. The oscillations on the b -spline are eliminated by adding energy cost functions including first and second derivatives. In these methods, the curve smoothness and fitting accuracy is controlled using several parameters, which are tuned by trial and error during approximation making it unfavorable to implement.

The real time interpolation of the $b$-splines has become an important research area in relation to curve fitting. Greenway and Zhang [92] first put in practice a b-spline curve motion interpolator for a six-axis robot based on Taylor's first-order expansion and compared the feed consistency to linear interpolators. Yang and Park [87] implemented a b-spline interpolator for wire electrical discharge machining (WEDM) also based on Taylor's first order expansion on a open architecture control system. Zhiming et. al. [93] used the first and second order Taylor's expansion with variable feedrate for real time b-spline interpolation. They adjusted the feedrate with respect to the curvature of the toolpath in order to consider the machine tool's dynamics. Tsai et. al. [74] compared several real time interpolation methods of b-splines such as the first and second order Taylor expansion, fourth order Runge - Kutta and an iterative parameter correction scheme. They indicated that second order .Taylor's expansion gives less feedrate command errors and requires relatively less computational power among the others. However, performing calculations for the necessary derivatives of $b$-splines is a computationally expensive duty within the servo system sampling interval. Later, Cheng et. al.[12] focused on this problem and proposed a more efficient $b$-spline evaluation structure for faster computation. The structure reduced the total number of computations for $b$-spline derivatives by half.

In this work, least squares approximation method is utilized to compress the G code blocks and to generate a smooth b-spline toolpath for five-axis machining. The toolpath is defined inverse kinematics independent enabling interpolation of the tool axis orientation vectors. Moreover, the generated tool path is smoothened, and its jerk content is reduced using linear and nonlinear optimization techniques to enhance the tracking performance of the servo system.

### 2.4. Trajectory Generation for 3-Axis Motions

Trajectory generation is vital for manufactured part quality. The generated trajectories must have continuous derivatives at least up to the second order to enable better tracking performance of the feed drive servo loop. Furthermore, acceleration and jerk profiles should have pre-specified limits in order to avoid saturating the actuators, or exciting unwanted dynamics in the servo or the mechanical structure. Various trajectory generation methods for 3 - Axis machining are summarized in the following.

Prithschow [59] brought up jerk limited trajectories with trapezoidal or sine square acceleration profiles for better tracking. Makino and Ohde [53] used a universal Cam curve for jerk limited trajectory. Later, Simon and Isik [68] also used trigonometric functions for the same purpose. Similarly, Macfarlane and Croft [52] designed a jerk limited trajectory generation method where they utilized a sine wave approximation to determine the velocity ramp from zero acceleration to non-zero acceleration for robots. Especially, for 3 - Axis machining, quintic spline [78, 79, 80] and b-spline trajectory generation [38] methods are used, which enable acceleration and jerk continuity. Altintas and Erkorkmaz [18] used a similar trapezoidal acceleration profile to generate online jerk limited trajectories on quintic spline toolpaths. Particularly, jerk limited trajectories have high smoothness making it easier to track but also avoid high frequency harmonics in the acceleration profile that would result in degrading the tracking and contouring accuracy. Same researchers also used second quintic spline in their work to further smoothen the reference position commands interpolated at varying time period and reconstructed it at servo loop frequency. Pritschow and Rogers [60] implemented the resampling as micro decentralized interpolation within the advanced digital feed drives. Later, Altintas and Erkorkmaz [20] developed a trajectory generation method with cubic acceleration profiles for high speed milling operations.

In this thesis, the fundamental requirements of the smooth trajectory such as acceleration and jerk continuity conditions are deployed to generate feedrate profiles for 5 - Axis machining. The research on generation trajectories for 5-Axis machine machining is presented with its critical issues in the following section.

### 2.5. Critical Issues on 5-Axis Spline Trajectory Generation

The implementation of such $\mathrm{C}^{2}$ and $\mathrm{C}^{3}$ jerk limited spline trajectory techniques into 5Axis is vital to adapt the process for HSM. Unlike to 3 - Axis, 5 - Axis machining centers have a non - cartesian kinematic structure $[7,64]$ that requires transformation of the tool tip position and tool axis orientations extracted by CAM systems into reference joint position commands using the inverse kinematics [51]. Some research has been done in robotic area on spline fitting and smooth trajectory generation techniques applied in the joint space [9, 39, 76]. Besides the robotic area, Chou and Yang [13, 14] presented an analytical study on the smooth command generation for 5 - Axis CNC machining by interpolating the joint angles with separate $\mathrm{n}^{\text {th }}$ order polynomials that are related to a single tool path parameter. In their work, they scheduled the curve parameter for smooth tool tip acceleration and constant velocity. Nevertheless, the kinematic limitations of machine tool's axes are not included in the study and can be violated because of tool path curvature and inverse kinematics transformation. Later, same researchers addressed that problem by using composite Hermite curves to generate piecewise constant velocity connected with smooth acceleration segments [15]. They included the machine tool's dynamic torque equation and axis jerk limitations to obtain the boundary conditions at the segment junctions and the maximum speed at the constant feed sections of the trajectory. Yang and Golub [85] also included a cutting forces predictor into the trajectory generation scheme and used inverse dynamics to enhance the tracking performance.

On the other hand, these methods require a machine dependent post processing step at CAM systems to generate offline joint splines that must be updated for each kinematic structure. A more general and efficient method is pursued to interpolate the toolpath position and tool orientation (See Fig. 2.5) regardless of the machine tool and surface type. The primary obstacle is the interpolation of the orientation vectors, while ensuring that the entire interpolated orientation spline remains on the unit sphere. The standard mathematical tools to interpolate the tool orientation are based on homogeneous transformation matrices, Euler angles [51], Frenet frames [77, 82], ruled [37] and implicit surfaces [83] as well as quaternions. For instance, Sirnivasan and Ge [71, 72] and Jutter [36] used quaternions to generate a dual three-space called the Image Space. They reflected the toolpath vectors into the image space and interpolated using rational Bezier and $b$-spline curves at $C^{3}$ continuity.

Cebula et. Al. [10] utilized the same idea using Nurbs and modified the orientation vectors in order to eliminate any singular location on the toolpath.


Figure 2.5: Tool Tip and Tool Orientation Vectors

To tackle the problem of tool orientation interpolation, Lin and Koren [40, 41] implemented the ruled surface technique on 5 - Axis machining. They have first extracted a path curve on the part surface and used the normal or the ruling of the surface directly as the tool orientation vector during point and flank milling operations along the curve. Since the surface is already expressed in a parametric form, the tool orientation can be computed continuously for each parameter in real time by the NC unit. The tool tip velocity is then held constant by the aid of first order Taylor expansion.

Spence et.al. [29] proposed a novel technique for smooth tool position and orientation interpolation using two nearly arc length parameterized quintic splines. The first quintic polynomial spline is used to interpolate the tool tip position vectors in cartesian coordinates. In addition to the tool tip spline, he defined a $C^{2}$ quintic Bezier curve on the spherical coordinates to be able to interpolate the unit tool axis orientation vectors (See Fig. 2.6). The spherical Bezier is also nearly arc length parameterized so that the constant tool rotation is achieved at equal curve parameter increment. He connected the travel distance to the tool orientation angle by the aid of a monotonic rational spline to achieve synchronized motion. This approach allowed generation of inverse kinematic independent trajectories for 5-Axis
machining where the NC unit only applies transformation to compute the joint angles during real time interpolation.


Figure 2.6: a) Position Spline --- b) Orientation Spline

Duc et. al. [47] addressed the tool orientation interpolation problem by using double bsplines defined practically in the workpiece coordinate system at the CAM stage. They presented the tool tip position vectors using a cubic $b$-spline. In addition, a second $b$-spline is employed in order to interpolate the position vectors of another point on the tool axis where the vector between the curves enables interpolation of continuous orientation vectors throughout the tool path. This method eliminates the need for a third curve definition for the coordination since both b-splines are parameterized by the same knot vector. Furthermore, the orientation of the tool can be modified using second $b$-spline to avoid singularities [1] or optimized [48] for flank milling operations. The double $b$-spline interpolation technique is also recognized by the Siemens NC units [69].

Parallel to the implementation of various spline interpolation techniques, development of feed optimal trajectory is necessary for five-axis high speed machining have gained high importance. It is a challenge to generate a smooth trajectory that enables the spline tool path to be traveled in minimum time without violating the physical limits of the axes drives. A significant amount of research has been done on time optimal trajectory generation for robots [6] and Cartesian machine tools [8]. However, those trajectories considered the maximum
torque or accelerations limitations resulting in a non-smooth trajectory that leads higher contouring errors. Constantinescu and Croft [16] used joint variables expressed as cubic polynomials for continuity. They optimized the cubic spline trajectory with respect to the torque rates for minimum cycle time. Results have showed enhanced tracking performance. On the other hand, the implementation of such optimization architecture is not adequate for machine tools' NC units because of its computational expense.

Weck et. al.[81] addressed that problem and developed a practical method to be used for machine tools processing spline toolpaths. He considered the worst curvature case in each spline segment to determine the maximum feedrate for that segment and connected the piecewise constant feed segments together with smooth accelerations transients. Although his method was easy to implement, higher velocities would be achieved in non-critical sections of the segments by using a continuously varying smooth feed profiles instead of piecewise constant.

If continuously varying feed profiles are preferred to use instead of piecewise constant velocity segments, the difficult task is obtaining a smooth velocity profile, which can tracked with less error by the drives. Flash and Hogan [28] first suggested minimum jerk trajectories for describing the smoothness of unconstrained human arm movements. Later, Kyriakopoulis and Saridis [44] proved that tracking performance is enhanced by minimizing the jerk content and implemented minimum jerk trajectories for robotic manipulators. Later they, focused on practically optimizing the trajectory for both minimum jerk and cycle time and minimized the maximum jerk value of the trajectory [45]. Piazzi and Visioli [57] also worked on minimizing the maximum magnitude of jerk using a global optimization algorithm for cubic trajectories by modulating the time durations of spline segments. In addition, Simon and Isik [68] used trigonometric splines for minimum jerk trajectory. They showed that the problem can be reduced into quadratic minimization problem, which makes it possible for practical implementation.

Altintas and Erkorkmaz [3] developed an efficient trajectory generation method for quintic spline tool paths, where the vitality of minimum jerk feed profile is combined with the cycle time objective. In order to schedule the feedrate continuously on the toolpath, they defined the displacement profile as a $\mathrm{C}^{3}$ quintic spline of time and optimized its coefficients for minimum jerk. The minimum cycle time objective is then injected by modulating the time
durations of each spline segment while respecting the machine tool's torque, velocity and jerk limits using an iterative algorithm. The algorithm is applied with a windowing technique for efficient processing and real time application.

A b-spline toolpath generation algorithm has developed in this thesis for 5-Axis machining based on the fitting a double spline toolpath on discrete data. The position and tool orientation errors are limited by user specified tolerances. Furthermore, a minimum jerk, minimum time trajectory is generated optimizing the quintic spline feed profile with respect to the machine tool's axis drive limits for the b-spline toolpath. The generated trajectory is then interpolated using the feed correction polynomial idea in real time.

## Chapter 3

## Spline Toolpath Generation from Discrete Toolpath Data for 5-Axis Machining

### 3.1. Introductionn

Reference trajectory generation plays a key role in the computer control of machine tools. Generated trajectories must have smooth kinematic profiles in order to maintain high tracking accuracy, and to avoid exciting the natural modes of the mechanical structure or servo control system. Especially for five-axis machining, the generation of a smooth trajectory is vital for the part quality.

For five-axis machining, after the design of the part, the primary machining toolpath is sampled for linear or circular interpolation by CAM software. For instance, these sampling algorithms extract relatively large number of points from the toolpath where the rate of change of curvature change is high and fewer points where the rate of change of curvature is low. This results in a G code containing thousands of polygonal segments to machine a sculptured surface. Feeding the machine tool with large number of discrete points leads to discontinuous movements of the axes for high speed machining since the line segmentation does not ensure continuity of the kinematic profiles. Therefore, spline trajectory generation techniques have become widely adopted in machining aerospace parts, dies, and molds. Their main advantage is that they provide more continuous feed motion than multiple linear or circular segments, and result in shorter machining time as well as better surface geometry.

The objective of the work presented in this chapter is to generate spline toolpaths for five-axis machine tools from initially sampled discrete toolpath data. The desired spline toolpaths are defined in terms of double b-splines [47]. The position and orientation of the tool can be interpolated directly using the toolpath format. Furthermore, two optimization techniques are presented in order to smooth the motion of a five-axis machine tool following the b -spline toolpath.

Herein, a b-spline toolpath fitting algorithm is developed by utilizing constrained least squares techniques in the toolpath generation part. It accepts discrete toolpath data consisting
of Tool Center Points (TCP) with tool orientations and generates two $b$-splines for a specified machining tolerance. The first b -spline is fitted to the translational component of the toolpath. It is used for the interpolation of the tool tip position. The second B-Spline fitted to the second set of points is used to interpolate the orientation component of the trajectory.

In the following optimization work, the quality of the b -spline tool path is improved. For smoother motion of the tool tip, the third derivative (i.e. Jerk) minimization objective is inserted into the least squares algorithm that delivers a smoother geometric profile of the tool tip to the machine tool. In addition, a nonlinear optimization algorithm is proposed for the upper $b$-spline path. This algorithm modifies the upper $b$-spline path by penalizing the angular jerk component of the toolpath. The resultant geometrical toolpath will enhance the tracking performance of the machine tool.


Figure 3.1: Flow Chart of Optimal B-spline Toolpath Generation
Firstly, a brief background is given for the $b$-spline representations and computations followed by the double $b$-spline polynomial representation of the toolpath in Section 3.2. The least squares based fitting of the double b-spline toolpath is presented in Section 3.3. The optimization algorithms applied on the spline toolpath are presented in the last section.

### 3.2. B-Spline Background

Cubic and quintic splines are being widely used for NC toolpath generation [78, 79, 80] as a replacement of line segmentation. For smoother trajectories, cubic polynomial splines are employed since they can ensure $C^{2}$ continuity allowing interpolation of acceleration continuous kinematic profiles. Similarly, quintic splines are utilized in the toolpath generation for a more consistent parameterization [78]. Unlike cubic and quintic polynomial splines, b-splines have only recently been adapted into the NC toolpath generation for machining because of their higher computational cost. However, $b$-splines overcome various deficiencies of previous splining techniques as follows:

- B-splines are compatible with the CAD Systems, since most parts are designed using surfaces generated by b-splines.
- B-splines are smoother curves.
- It is easier to modify and predict the shape of the curve with the aid of control points.
- The curve can be modified locally without affecting the shape of the complete spline.

A b-spline is illustrated in Fig. 3.2. b-splines are a generalization of Bezier curves. They are defined by a control polygon consisting of $n+1$ control points $P_{i}, \quad i=0, \ldots n$ that form the shape of the curve.


Figure 3.2: B-spline Toolpath
By definition, a $\mathrm{p}^{\text {th }}$ degree b -spline constructed between points $P_{0}$ and $P_{n}$ is parameterized in the interval of $\left[\begin{array}{ll}0 & 1\end{array}\right]$ by a knot vector, $U$. The knot vector contains $m=n+p+1$ knots $(\mu)$ that are initially allocated spline parameters $(u)$ for certain positions on the $b$-spline (See Fig. 3.2). The knot vector of the $b$-spline is given as:

$$
U=[\underbrace{\mu_{0}, \ldots, \mu_{p}}_{\text {Repeitive } \mathrm{p}+1 \text { Knols }}, \mu_{p+1}, \mu_{p+2}, \ldots \mu_{j}, \ldots, \mu_{n-1}, \mu_{n}, \underbrace{\mu_{n+1}, \ldots, \mu_{n+p+1}}_{\text {Repetitive } \mathrm{p}+1 \text { Knots }}] \text {, }
$$

where

$$
\mu_{j} \leq \mu_{j+1}, j=0,1, \ldots, m, \quad \mu \in\left[\begin{array}{ll}
0 & 1 \tag{3.1}
\end{array}\right] .
$$

The consecutive knots [ $\mu_{j} \mu_{j+1}$ ] in the knot vector are called as knot spans. In addition, the knot vector contains $p+1$ repeated knots at the beginning and end of the knot vector in
order to anchor the $b$-spline at those control points. For instance, the knot vector of a cubic $(p=3) b$-spline is given as:

$$
U=\left[0,0,0,0, \mu_{p+1}, \mu_{p+2}, \ldots \mu_{j}, \ldots, \mu_{n-1}, \mu_{n}, 1,1,1,1\right] .
$$

The evaluation of $b$-spline is a computationally expensive. Any position on cubic or quintic splines are expressed by a single explicit polynomial equation such as:

$$
\begin{aligned}
& S_{x}(u)=a_{x, n} u^{n}+a_{x, n-1} u^{n-1}+\ldots+a_{x, 0} u^{0} \\
& S_{y}(u)=a_{y, n} u^{n}+a_{y, n-1} u^{n-1}+\ldots+a_{y, 0} u^{0} .
\end{aligned}
$$

However, since b-splines are built up by intermittent piecewise polynomials (basis functions) that blend the control points. A recursive algorithm is used in order to compute the basis functions. The $\mathrm{i}^{\text {th }} \mathrm{b}$-spline basis function of degree $p$ is denoted by $N_{i, p}$. For example a zero degree basis functions is defined as:

$$
N_{i, 0}(u)=\left\{\begin{array}{l}
1 \text { if } \mu_{i} \leq u \leq \mu_{i+1}  \tag{3.2}\\
0 \quad \text { otherwise } .
\end{array}\right.
$$

As indicated in Eq.(3.2), zero degree basis functions are step functions. Higher degree basis functions $\left(N_{i, 1}, N_{i, 2}, \ldots, N_{i, p}\right)$ are linear combinations of two lower degree basis functions expressed according to the following recursive scheme:

$$
\begin{align*}
& N_{i, 1}(u)=\frac{u-\mu_{i}}{\mu_{i+1}-\mu_{i}} N_{i, 0}(u)+\frac{\mu_{i+p+1}-u}{\mu_{i+p+1}-\mu_{i+1}} N_{i+1,1}(u) \\
& N_{i, 2}(u)=\frac{u-\mu_{i}}{\mu_{i+1}-\mu_{i}} N_{i, 1}(u)+\frac{\mu_{i+p+1}-u}{\mu_{i+p+1}-\mu_{i+1}} N_{i+1,2}(u) \\
& \vdots  \tag{3.3}\\
& N_{i, p-1}(u)=\frac{u-\mu_{i}}{\mu_{i+1}-\mu_{i}} N_{i, p-2}(u)+\frac{\mu_{i+p+1}-u}{\mu_{i+p+1}-\mu_{i+1}} N_{i+1, p-2}(u) \\
& \Rightarrow N_{i, p}(u)=\frac{u-\mu_{i}}{\mu_{i+1}-\mu_{i}} N_{i, p-1}(u)+\frac{\mu_{i+p+1}-u}{\mu_{i+p+1}-\mu_{i+1}} N_{i+1, p-1}(u)
\end{align*}
$$

and various basis functions are illustrated in Fig. 3.3.
a) Step Basis Functions

b) $1^{\text {st }}$ Degree Basis Functions

c) $2^{\text {nd }}$ Degree Basis Functions


Figure 3.3: B-spline Basis Functions

Once the non-zero basis functions are computed for the spline parameter $(u)$, a position on the $b$-spline is obtained by the sum of only the $p^{\text {th }}$ degree basis functions and control points as: Machining

$$
\left.\begin{array}{rl}
P(u) & =N_{0, p}(u) \cdot P_{0}+N_{1, p}(u) \cdot P_{1}+\ldots+N_{n-1, p}(u) \cdot P_{n-1}+N_{n, p}(u) \cdot P_{n} \\
& =\sum_{0}^{n} N_{i, p}(u) P_{i} \tag{3.4}
\end{array}\right\}
$$

Similarly, derivatives of the $b$-spline curves are calculated using the derivatives of basis functions. The first derivative of the $\mathrm{p}^{\text {th }}$ degree basis function with respect to the curve parameter is given by a similar recursive scheme:

$$
\begin{equation*}
\frac{\mathrm{d} N_{i, p}(u)}{\mathrm{d} u}=\frac{p}{\mu_{i+p}-\mu_{i}} N_{i, p-1}(u)-\frac{p}{\mu_{i+p+1}-\mu_{i+1}} N_{i+1, p-1}(u), \tag{3.5}
\end{equation*}
$$

and the $\mathrm{k}^{\text {th }}$ derivative of the basis functions is expressed as:

$$
\begin{equation*}
\frac{\mathrm{d}^{(\mathrm{k})} N_{i, p}(u)}{\mathrm{d} u^{(k)}}=p\left(\frac{\mathrm{~d}^{(k-1)} N_{i, p-1}(u) / \mathrm{d} u^{(k-1)}}{\mu_{i+p}-\mu_{i}}-\frac{\mathrm{d}^{(k-1)} N_{i+1, p-1}(u) / \mathrm{d} u^{(k-1)}}{\mu_{i+p+1}-\mu_{i+1}}\right) . \tag{3.6}
\end{equation*}
$$

Then, the derivative of the $b$-spline is obtained by:

$$
\left.\begin{array}{rl}
\frac{\mathrm{d}^{(k)} P(u)}{\mathrm{d} u^{(k)}} & =\frac{\mathrm{d}^{(k)} N_{0, p}(u)}{\mathrm{d} u^{(k)}} \cdot P_{0}+\frac{\mathrm{d}^{(k)} N_{\mathrm{t}, p}(u)}{\mathrm{d} u^{(k)}} \cdot P_{1}+\ldots+\frac{\mathrm{d}^{(\mathrm{k})} N_{n-1, p}(u)}{\mathrm{d} u^{(k)}} \cdot P_{n-1}+\frac{\mathrm{d}^{(\mathrm{k})} N_{n, p}(u)}{\mathrm{d} u^{(k)}} \cdot P_{n} \\
& =\sum_{0}^{n} \frac{\mathrm{~d}^{(k)} N_{i, p}(u)}{\mathrm{d} u^{(k)}} \cdot P_{i} . \tag{3.7}
\end{array}\right\}
$$

### 3.3. Double B-Spline Polynomial Representation

A toolpath consists of the tool tip position and associated orientations of the tool axis in five-axis machining. In order to maintain this information of the tool motion, a spline toolpath format is put forward by Lartigue at. al.[47]. The proposed toolpath consists of two splines, which are dedicated for the following:

- The locus of the tool tip position $\left(x^{t}, y^{t}, z^{\prime}\right)$ is given by a spline, $P(u)$ in part coordinates.
- Another point on the tool axis $\left(x^{h}, y^{h}, z^{h}\right)$ is represented by a second spline, $Q(u)$.


Figure 3.4: Double B-Spline Toolpath Representation

The overall representation of the spline toolpath is illustrated in Fig. 3.4. The lower spline $P(u)$ is used to express only the tool tip position as in regular 3-Axis machining and the upper spline $Q(u)$ interpolates another point on the tool axis.

The relative distance between $P(u)$ and $Q(u)$ is defined as $H$, which can be defined depending on the machining operation such as flank or point milling. For flank milling, the relative distance refers to the metal cutting contact height that corresponds to the active part of the tool or depth of cut. Similarly, for point milling $H$ is the diameter of the tool's ballend as illustrated in Fig. 3.5.


Figure 3.5: Upper and Lower B-splines

The splines of the toolpath, $P(u)$ and $Q(u)$ are given in non-uniform b-spline form. The expression of lower and upper b-splines defined on the same knot vector, $(U)$ is expressed as:

$$
\begin{align*}
& P(u)=\left[\begin{array}{l}
x^{t}(u) \\
y^{t}(u) \\
z^{t}(u)
\end{array}\right]=\sum_{i=0}^{n} N_{i, p}(u) P_{i}=\sum_{i=0}^{n} N_{i, p}(u) \cdot\left[\begin{array}{l}
P_{x} \\
P_{y} \\
P_{z}
\end{array}\right], \quad u \in U \\
& Q(u)=\left[\begin{array}{l}
x^{h}(u) \\
y^{h}(u) \\
z^{h}(u)
\end{array}\right]=\sum_{i=0}^{n} N_{i, p}(u) Q_{i}=\sum_{i=0}^{n} N_{i, p}(u) \cdot\left[\begin{array}{l}
Q_{x} \\
Q_{y} \\
Q_{z}
\end{array}\right], \quad u \in U \tag{3.8}
\end{align*}
$$

where $P_{i}=\left[\begin{array}{lll}P_{x} & P_{y} & P_{z}\end{array}\right]^{T}$ and $Q_{i}=\left[\begin{array}{lll}Q_{x} & Q_{y} & Q_{z}\end{array}\right]^{T}, i=0, \ldots, n$ are the $n+1$ set control points of the lower and upper b-splines, respectively. The degree of the $b$-splines is selected either $p=3$ or higher depending on the continuity requirement. For instance, the cubic double B-Spline toolpath allows definition of a $\left(C^{2}\right)$ continuous motion. Similarly, a $5^{\text {th }}$ degree (quintic) b-spline toolpath enables interpolation of jerk continuous $\left(C^{4}\right)$ motion.

The tool tip position is directly interpolated by $P(u)$. However, in order to interpolate the tool orientation, both b-splines are utilized. As shown in Fig. 3.5, the vector $\vec{A}(u)$ defines the Machining
tool axis. It can be expressed for each spline parameter simply by subtracting the lower bspline from the upper one as:

$$
A(u)=Q(u)-P(u)=\left[\begin{array}{l}
x^{h}-x^{t}  \tag{3.9}\\
y^{h}-y^{t} \\
z^{h}-z^{t}
\end{array}\right]=\left[\begin{array}{l}
A_{x} \\
A_{y} \\
A_{z}
\end{array}\right]=A_{x} \dot{i}+A_{y} \vec{j}+A_{z} \vec{k} .
$$

The unit orientation vector of the tool axis, $O(u)$ is computed similarly when $A(u)$ is normalized by the relative distance between $P(u)$ and $Q(u)$ as:

$$
O(u)=\left[\begin{array}{l}
O_{x}  \tag{3.10}\\
O_{y} \\
O_{z}
\end{array}\right]=O_{x} \dot{i}+O_{y} \vec{j}+O_{z} \vec{k}=\frac{A_{x}(u)}{|\vec{A}|} \dot{i}+\frac{A_{y}(u)}{|\vec{A}|} \vec{j}+\frac{A_{z}(u)}{|\vec{A}|} \vec{k}
$$

where $O(u)$ is also called orientation spline of the trajectory. It is expressed in terms of the upper and lower $b$-spline control points by:

$$
\begin{equation*}
O(u)=\frac{\sum_{i=0}^{n} N_{i, p}(u) \cdot\left(Q_{i}-P_{i}\right)}{\left\|\sum_{i=0}^{n} N_{i, p}(u) \cdot\left(Q_{i}-P_{i}\right)\right\|}, u \in U . \tag{3.11}
\end{equation*}
$$

Hence, the objective of continuously expressing the tool axis orientation and tool tip position is achieved during simultaneous interpolation of both splines using Eq. (3.8).

### 3.4. Generation of Lower B-Spline

The tool tip positions are initially given as a batch of discrete tool center points (TCP) by CAM systems. The fitting algorithm for generation of the lower b-spline through TCP data is presented in this section. For 5 - Axis machining of free-form surfaces, large number of TCPs is needed and interpolating all the data may result in a non-smooth spline. In order to
generate a smooth b-spline toolpath, a least squares approximation is used instead of the exact interpolation. A tolerance band is defined to limit the least squares fitting errors to respect the machining tolerances. Furthermore, the algorithm is extended as a general least squares to impose any specific position, tangent and higher derivative constraints onto the toolpath.

The lower b-spline is already defined in Eq.(3.8) as,

$$
P(u)=\left[\begin{array}{l}
x^{t}(u) \\
y^{t}(u) \\
z^{t}(u)
\end{array}\right]=\sum_{i=0}^{n} N_{i, p}(u) P_{i}=\sum_{i=0}^{n} N_{i, p}(u) \cdot\left[\begin{array}{l}
P_{x} \\
P_{y} \\
P_{z}
\end{array}\right], \quad u \in U,
$$

and the above expression shows that modification of a control point affects only the associated axes components of the b-spline. Therefore, the least squares can be applied to each axis component separately. The formulation used for fitting the $x^{t}$ component of the lower $b$-spline is presented here and same formulation is then applied to the rest of the $b$ spline components.


Figure 3.6: Initial Spline Parameter Calculation

The tool tip position $\left(q_{k}^{t}\right)$ is given by $M+1$ data points, $q_{0}^{t}, q_{1}^{t}, \ldots, q_{k}^{t}, q_{M}^{t}$ where $q_{k}^{t}=\left[\begin{array}{lll}q_{x_{k}}^{t} & q_{y_{k}}^{t} & q_{z_{k}}^{t}\end{array}\right]^{T}$. The least squares algorithm starts with seeking the best prediction, $\hat{q}_{x_{k}}^{\prime}$ for any tool tip point $q_{x_{k}}^{t}$ at $\left.x^{t}(u)\right|_{u=\bar{u}_{k}}$ on the lower b-spline,

$$
\hat{q}=\left[\begin{array}{lllll}
\hat{q}_{x_{0}}^{t} & \hat{q}_{x_{1}}^{t} & \hat{q}_{x_{2}}^{t} & \ldots & \hat{q}_{x_{H}}^{t} \tag{3.12}
\end{array}\right]^{T} \Leftrightarrow x^{t}(u)=N_{0, p}(u) \cdot P_{x 0}+N_{1, p}(u) \cdot P_{x 1}+\ldots+N_{n, p}(u) \cdot P_{x n}
$$

where $u=\left[\begin{array}{lllll}\bar{u}_{0} & \bar{u}_{1} & \bar{u}_{2} & \ldots & \bar{u}_{M}\end{array}\right]$ are the parameter values on the b -spline corresponding to the given tool center points $\left(q_{k}^{t}\right)$ shown in Fig. 3.6. The chord length parameterization method [58] is applied to calculate the initial spline parameters. This method is used for most of the data distributions and gives a rough estimation of the spline parameters while only considering the Euclidian distance between the points.

The total cord length between the given data points is defined as,

$$
\begin{equation*}
S_{C}=\sum_{k=1}^{t}\left|q_{k}^{t}-q_{k-1}^{t}\right|=\sum_{k=1}^{M} \sqrt{\left(q_{x_{k}}^{t}-q_{x_{k-1}}^{t}\right)^{2}+\left(q_{y_{k}}^{t}-q_{y_{k-1}}^{t}\right)^{2}+\left(q_{z_{k}}^{t}-q_{z_{k-1}}^{t}\right)^{2}} . \tag{3.13}
\end{equation*}
$$

The initial and final parameter values are set as $\bar{u}_{0}=0, \bar{u}_{M}=1$ and the internal spline parameters are allocated proportional to the consecutive chord lengths as:

$$
\begin{equation*}
\bar{u}_{k}=\bar{u}_{k-1}+\frac{\left|q_{k}^{t}-q_{k-1}^{t}\right|}{S_{C}} \quad k=1, \ldots M-1 \tag{3.14}
\end{equation*}
$$

Thereafter, the knot vector, $(U)$ of the b -spline is required for the parameterization and setting up the equations for the least squares fitting. The initial and final repetitive knots in the knot vector are defined by Eq.(3.1) initially as:

$$
\mu_{0}=\ldots=\mu_{0}=0 \quad \text { and } \quad \mu_{n+1}=\ldots=\mu_{n+p+1}=1
$$

The remaining knots are obtained as described by Piegl [58]. Defining $d$ as a positive real number, denoted by $i=\operatorname{int}(d)$ the largest integer such that $i \leq d$,

$$
\begin{equation*}
d=\frac{m+1}{n-p+1} \tag{3.15}
\end{equation*}
$$

where $n$ is the number of control points, $p$ is the degree of the b -spline and $m=n+p+1$. The internal knots in the knot vector are calculated by:

$$
\left.\begin{array}{l}
i=\operatorname{int}(j \cdot d)  \tag{3.16}\\
\alpha=j \cdot d-i \\
\mu_{p+j}=(1-\alpha) \bar{u}_{i-1}+\alpha \bar{u}_{i} \quad j=1, \ldots, n-p
\end{array}\right\} .
$$

Using Eq.(3.16) and Eq.(3.1) the knot vector of the spline toolpath is formed. Corresponding position predictions for the data points are obtained on the b-spline using Eq. (3.14), which results in the following system of equations:

$$
\hat{q}_{x}^{t}=\left[\begin{array}{c}
\hat{q}_{x_{0}}^{t}  \tag{3.17}\\
\hat{q}_{x_{1}}^{t} \\
\hat{q}_{x_{2}}^{t} \\
\vdots \\
\hat{q}_{x_{M}}^{t}
\end{array}\right]=\underbrace{\left[\begin{array}{ccccc}
N_{0, p}\left(\bar{u}_{0}\right) & N_{1, p}\left(\bar{u}_{0}\right) & N_{2, p}\left(\bar{u}_{0}\right) & \ldots & N_{n, p}\left(\bar{u}_{0}\right) \\
N_{0, p}\left(\bar{u}_{1}\right) & N_{1, p}\left(\bar{u}_{1}\right) & N_{2, p}\left(\bar{u}_{1}\right) & \ldots & N_{n, p}\left(\bar{u}_{1}\right) \\
N_{0, p}\left(\bar{u}_{2}\right) & N_{1, p}\left(\bar{u}_{2}\right) & N_{2, p}\left(\bar{u}_{2}\right) & \ldots & N_{n, p}\left(\bar{u}_{2}\right) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
N_{0, p}\left(\bar{u}_{M}\right) & N_{1, p}\left(\bar{u}_{M}\right) & N_{2, p}\left(\bar{u}_{M}\right) & \ldots & N_{n, p}\left(\bar{u}_{M}\right)
\end{array}\right]}_{\boldsymbol{D}} \underbrace{\left[\begin{array}{c}
P_{x_{0}} \\
P_{x_{1}} \\
P_{x_{2}} \\
\vdots \\
P_{x_{n}}
\end{array}\right]}_{P_{x}}
$$

where $\Phi$ is a $M+1 \times n+1$ matrix consist of basis functions. $\left(N_{i, p}(u)\right)$ evaluated at each $\bar{u}_{k}$, and $P_{x}$ is the unknown control point vector. The errors (residues) of the predictions are defined as the deviation of the approximated points $\hat{q}_{x_{k}}^{t}=x^{t}(u)_{u=\bar{u}_{k}}$ on the b-spline from the actual tool tip positions, $q_{x_{k}}^{t}$ expressed as:

$$
\begin{equation*}
e_{x}=q_{x}^{t}-\hat{q}_{x}^{t}=q_{x}^{t}-\sum_{i=0}^{n} N_{i, p}(\bar{u}) P_{x i}=q_{x}^{t}-\Phi P_{x} . \tag{3.18}
\end{equation*}
$$

The objective function of the least squares to be minimized is then;

$$
\begin{equation*}
J_{L_{x}}=\frac{1}{2} e_{x}^{T} e_{x}=\frac{1}{2}\left(q_{x}^{t}-\sum_{i=0}^{n} N_{i, p}(\bar{u}) P_{x i}\right)^{T}\left(q_{x}^{t}-\sum_{i=0}^{n} N_{i, p}(\bar{u}) P_{x i}\right)=\frac{1}{2}\left(q_{x}^{t}-\Phi P_{x}\right)^{T}\left(q_{x}^{t}-\Phi P_{x}\right) . \tag{3.19}
\end{equation*}
$$

The formulation of least squares implies that the derivative of the objective function with respect to the control points $P_{x}$ is set to zero.

$$
\left.\begin{array}{l}
\frac{\partial J_{L}}{\partial P}=-\Phi^{T}\left(q_{x}^{t}-\Phi P_{x}\right) \\
\Phi^{T} \Phi P_{x}=\Phi q_{x}^{t}  \tag{3.20}\\
P_{x}=\left(\Phi^{T} \Phi\right)^{-1} \Phi q_{x}^{t}
\end{array}\right\}
$$

Eq.(3.20) gives the control points for a regular least squares fit. The $b$-spline can also be constrained at some certain points. For instance, position constraints at tool tip points can be introduced in order to eliminate any fitting errors. In addition, tangents or possibly higher derivatives required to be imposed on the curve. In order to enforce the desired shape constraints, a general linear system of equations is composed in the constrained least squares algorithm.

For the specific position constraints $\left.x^{t}(u)\right|_{u=\bar{u}_{k}}=\xi_{k}^{(0)}, k=0 \ldots l_{0}$, linear equations are grouped as follows:

$$
\underbrace{\left[\begin{array}{c}
\xi_{0}^{(0)}  \tag{3.21}\\
\vdots \\
\xi_{l_{0}}^{(0)}
\end{array}\right]}_{\xi^{(0)}}=\underbrace{\left[\begin{array}{ccccc}
N_{0, p}\left(\bar{u}_{0}\right) & N_{1, p}\left(\bar{u}_{0}\right) & N_{2, p}\left(\bar{u}_{0}\right) & \ldots & N_{n, p}\left(\bar{u}_{0}\right) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
N_{0, p}\left(\bar{u}_{l_{0}}\right) & N_{1, p}\left(\bar{u}_{l_{0}}\right) & N_{2, p}\left(\bar{u}_{l_{0}}\right) & \ldots & N_{n, p}\left(\bar{u}_{l_{0}}\right)
\end{array}\right]}_{L^{(0)}} \cdot \underbrace{\left[\begin{array}{c}
P_{x_{0}} \\
P_{x_{1}} \\
P_{x_{2}} \\
\vdots \\
P_{x_{n}}
\end{array}\right] . . ~}_{P_{x}}
$$

Tangent, $\left.\quad \frac{\mathrm{d} x^{t}(u)}{\mathrm{d} u}\right|_{u=\bar{u}_{k}}=\xi_{k}^{(1)}, \quad k=0 \ldots l_{1 .} \quad$ and $\quad$ normal $\left.\quad \frac{\mathrm{d}^{2} x^{t}(u)}{\mathrm{d} u^{2}}\right|_{u=\bar{u}_{k}}=\xi_{k}^{(2)}, \quad k=0 \ldots l_{2}$ constraints can be further imposed in the following matrix equations (Eq.(3.7)) by using the derivatives of the basis functions given as $\quad N_{i, p u}\left(\bar{u}_{k}\right)=\left.\frac{\mathrm{d} N_{i, p}}{\mathrm{~d} u}\right|_{u=\bar{u}_{k}}$ and $N_{i, p u u}\left(\bar{u}_{k}\right)=\left.\frac{\mathrm{d}^{2} N_{i, p}}{\mathrm{~d} u^{2}}\right|_{u=\bar{u}_{k}}$, which are calculated by Eq. (3.6),

The resultant constraint matrix is $L=\left[\begin{array}{l}L^{(0)} \\ L^{(1)} \\ L^{(2)}\end{array}\right]_{r \times n+1}$, where $r=\left(l_{0}+1\right)+\left(l_{1}+1\right)+\left(l_{2}+1\right)$ and the overall constraint vector containing position and derivative constraints is $\xi=\left[\begin{array}{l}\xi^{(0)} \\ \xi^{(1)} \\ \xi^{(2)}\end{array}\right]_{r \times 1}$. Hence, the lower b-spline is obtained by solving the following optimization problem:

$$
\begin{equation*}
J_{L}=\min _{P_{x}^{t}} \frac{1}{2}\left(q_{x}^{t}-\Phi P_{x}\right)^{T}\left(q_{x}^{t}-\Phi P_{x}\right) \quad \text { Subject to: } \quad L \cdot P_{x}=\xi \tag{3.23}
\end{equation*}
$$

which is a linear quadratic minimization problem, and can be solved using Lagrange Multipliers method. Introducing the Lagrange multipliers: $\Lambda=\left[\begin{array}{lllll}\Delta_{0} & \Delta_{1} & \Delta_{2} & \cdots & \Delta_{r}\end{array}\right]$, the augmented objective function is written as:

$$
\begin{equation*}
\bar{J}_{L_{x}}(P, \Lambda)=\frac{1}{2}\left(q_{x}^{t}-\Phi P_{x}\right)^{T}\left(q_{x}^{t}-\Phi P_{x}\right)+\Lambda\left(L \cdot P_{x}-\xi\right) \tag{3.24}
\end{equation*}
$$

Equating the partial derivatives to zero $\left(\frac{\partial \bar{J}_{L}}{\partial P}=0\right.$ and $\left.\frac{\partial \bar{J}_{L}}{\partial \Lambda}=0\right)$ yields the linear equation system:

$$
\left.\begin{array}{l}
\Phi^{T} \Phi P_{x}+L^{T} \Lambda=\Phi^{T} q_{x}^{t}  \tag{3.25}\\
L \cdot P_{x}=\xi
\end{array}\right\} \Rightarrow\left[\begin{array}{cc}
\Phi^{T} \Phi & L^{T} \\
L & 0
\end{array}\right] \cdot\left[\begin{array}{c}
P_{x} \\
\Lambda
\end{array}\right]=\left[\begin{array}{c}
\Phi^{T} q_{x}^{t} \\
\xi
\end{array}\right] .
$$

From Eq.(3.25), a solution can be obtained only if $\Phi \cdot \Phi^{T}$ is positive-definite and nonsingular. Every knot span, $\left[\mu_{j}, \mu_{j+1}\right.$ ] must contain one $\bar{u}$ to satisfy the non-singularity condition of $\Phi \cdot \Phi^{T}$. Eq.(3.16) guarantees that every knot span contains at least one $\bar{u}$, which eliminates the possibility of singularity. Furthermore, provided that the constraints are linearly independent and less than the unknowns $(r \leq n)$, a unique solution is obtained for the constrained least squares fit.

### 3.4.1. Tolerance Definition and Parameter Correction

The general least square algorithms are simply based on minimizing the total vertical errors in each axis component of the b-spline. However, in this application, it must be ensured that the approximated b-spline stays within a specified accuracy, namely the machining tolerance. A fitting error limit $\left(e_{\text {max }}\right)$ is defined to control deviation (e) on the lower b-spline from the reference tool center points. In Fig. 3.7, the tool center points, the proposed $b$-spline fit and the definition of the tolerance band to limit the errors can be seen. The tolerance band is defined as a pipe with a radius of $\left(e_{\max }\right)$ in 3 dimensional space covering the fitted b -spline.


Figure 3.7: Tolerance Definition for Lower B-spline Generation

The deviation on the lower b-spline from every tool center point is required to check the errors. Corresponding spline parameters $(\bar{u})$ to the tool center points $\left(q^{t}\right)$ are already obtained by the chord length approximation. However, as shown in Fig. 3.8, this approximation does not ensure the correct spline parameter for each tool center point on the generated b-spline. The closest spline parameters must be identified in order to calculate accurate norm of deviation from given points, which is done by scanning throughout $b$-spline for the correct parameter.

The vector between the lower b-spline $P=\left[\begin{array}{lll}x^{t} & y^{t} & z^{t}\end{array}\right]^{T}$ and the given tool tip point $q^{t}=\left[\begin{array}{lll}q_{x}^{t} & q_{y}^{t} & q_{z}^{t}\end{array}\right]^{T}$ is written as:

$$
\begin{equation*}
\vec{D}=\left(x^{t}-q_{x}^{t}\right) \dot{j}+\left(y^{t}-q_{y}^{t}\right) \vec{j}+\left(z^{t}-q_{z}^{t}\right) \vec{k} . \tag{3.26}
\end{equation*}
$$ Machining

It is known that the vector $\vec{D}$ must be perpendicular to tangent of the spline at $u$ for the closest point. A dot product function is formed to ensure the condition as shown in Fig. 3.8;

$$
\begin{equation*}
f(u)=\frac{\mathrm{d} P(u)}{\mathrm{d} u} \cdot\left(P(u)-q^{t}\right) \tag{3.27}
\end{equation*}
$$



Figure 3.8: Spline Parameter Correction

The root of the Eq. (3.27) is found by Newton Raphson iteration. Parameter $u_{i}$ is obtained by the $i^{\text {th }}$ iteration,

$$
\begin{equation*}
u_{i+1}=u_{i}-\frac{f(u)}{\frac{\mathrm{d} f(u)}{\mathrm{d} u}}=u_{i}-\left.\frac{\frac{\mathrm{d} P(u)}{\mathrm{d} u}\left(P(u)-q^{t}\right)}{\frac{\mathrm{d}^{2} P(u)}{\mathrm{d} u^{2}}\left(P(u)-q^{t}\right)+|\mathrm{d} P(u) / \mathrm{d} u|^{2}}\right|_{u=u_{i}} \tag{3.28}
\end{equation*}
$$

The convergence of the iteration in Eq. (3.28) to the correct spline parameter depends on the initial guess. A very close initial guess is simply the $\bar{u}$ values computed initially by the chord length approximation by Eq. (3.14). Two zero tolerances, shown on Fig. 3.9, are defined to indicate the convergence:
$\varepsilon_{1}$ is the measure of the Euclidean distance and $\varepsilon_{2}$ is to indicate zero cosine value.


Figure 3.9: Zero Tolerance Definitions a) Zero Cosine --- b) Point Coincidence

Using the above zero tolerances, the criteria to terminate the iteration is as follows:
(1) if $\left|P(u)_{u=u_{i}}-q^{t}\right| \leq \varepsilon_{1}$
the point lies on the b-spline at the calculated parameter (point coincidence) or,
(2) if $\left.\frac{\frac{\mathrm{d} P(u)}{\mathrm{d} u}\left(P(u)-q^{t}\right)}{\left.\left|\frac{\mathrm{d} P(u)}{\mathrm{d} u}\right|\left(P(u)-q^{t}\right) \right\rvert\,}\right|_{u=u_{1}} \leq \varepsilon_{2}$
the vector $\vec{D}$ is the normal of the curve at the spline parameter and condition for Eq.(3.27) is satisfied or,

$$
\begin{equation*}
\text { if } \left.\left|\left(u_{i+1}-u_{i}\right) \frac{\mathrm{d} P(u)}{\mathrm{d} u}\right|_{u=u_{i}} \right\rvert\, \leq \varepsilon_{1} \tag{3}
\end{equation*}
$$

the spline parameter does not change significantly during the Newton Raphson iteration (Eq. (3.28)).

For basic machining operations, $\varepsilon_{1}$ in Eq. (3.29) and Eq. (3.31) can be taken as $10^{-6}[\mathrm{~mm}]$ to indicate that the point lies on the b -spline at the searched spline parameter. On the other hand, for precision machining this threshold value should be selected more conservatively. The zero cosine measure, $\varepsilon_{2}$ in Eq.(3.30) is selected less than $\varepsilon_{1}$ such as $10^{-8}$.

If any of the above conditions is satisfied (Eq. (3.29), Eq. (3.30) or Eq. (3.31)), the iteration is terminated and correct norm of the error is found at $u_{i}$ as:

$$
\begin{equation*}
e_{k}=|\vec{D}|_{u=u_{i}}=\sqrt{\left(x^{t}\left(u_{i}\right)-q_{x_{k}}^{t}\right)^{2}+\left(y^{t}\left(u_{i}\right)-q_{y_{k}}^{t}\right)^{2}+\left(z^{t}\left(u_{i}\right)-q_{z_{k}}^{t}\right)^{2}} . \tag{3.32}
\end{equation*}
$$

Increasing the number of control points, $n$ improves the control on the spline shape. Eventually, the maximum norm of the error can be reduced to the desired value by increasing the total number of control points in an iterative approach during least squares fit. The procedure is shown in Fig. 3.10.


Figure 3.10: Lower B-spline Fitting Algorithm

The fitting process of the lower b-spline starts with less number of control points. The number of initial control points can be judged for free-form surface machining as $5[\%]$ of the total given TCP $(n=(M+1) / 20)$. Using Eq. (3.1), Eq. (3.16), the knot vector is generated and the lower b-spline is fitted using Eq. (3.25). Following that, resultant errors are checked using Eq. (3.27) and Eq. (3.32). The knot span, where the maximum tolerance violation occurs, is recorded as un-converged. An extra knot, $\hat{\mu}$ is added in the middle of the unconverged knot span by averaging the neighbouring knots as:

$$
\begin{equation*}
\hat{\mu}_{j+1}=\frac{\mu_{j}+\mu_{j+1}}{2} . \tag{3.33}
\end{equation*}
$$

Adding extra knots only into the specific knot spans will assist reducing the errors focused mainly at this portion of the $b$-spline. The $b$-spline is then fitted again with the new knot vector using Eq. (3.25). Errors are checked after each fit until all the knot spans are converged and the max error is kept under given limit.

### 3.5. Generation of the Upper Spline

The fitting of $P(u)$ through the set of tool center points in the previous section. Similarly, a second set of points on the tool axis is required to generate the upper b-spline $Q(u)$. Firstly, the reference tool center points are updated on the lower b-spline using spline parameters obtained from Eq. (3.28). The updated points are offset in the direction of reference orientation vectors $O^{r}=\left[\begin{array}{lll}O_{x}^{r} & O_{y}^{r} & O_{z}^{r}\end{array}\right]^{\mathrm{T}}$ by the amount of $H$ as,

$$
\begin{equation*}
q_{k}^{h}=P\left(u_{k}\right)+H \cdot O_{k}^{r} \quad k=0, \ldots, M \tag{3.34}
\end{equation*}
$$

where $q^{h}$ is the new set of reference points in cartesian coordinates for another location on the tool axis. The upper b-spline $Q(u)$ is then fitted through $q^{h}$ using Eq. (3.25) while checking only for the orientation errors $e_{\theta}$. The orientation error is defined as angle between given reference orientation $O^{r}$ and the approximated orientation, $O$ as:

$$
\begin{align*}
& \sin \left(e_{\theta_{k}}\right)=\frac{\left|O_{k}^{r} \times O_{k}\right|}{\left|O_{k}^{r}\right|\left|O_{k}\right|}, \quad \cos \left(e_{\theta_{k}}\right)=\frac{O_{k}^{r} \cdot O_{k}}{\left|O_{k}^{r}\right| O_{k} \mid} \\
& \left.e_{\theta_{k}}=\tan ^{-1}\left(\frac{\sin \left(e_{\theta_{k}}\right)}{\cos \left(e_{\theta_{k}}\right)}\right)=\tan ^{-1}\left(\frac{\left|O_{k}^{r} \times O_{k}\right|}{O_{k}^{r} \cdot O_{k}}\right)\right\} . \tag{3.35}
\end{align*}
$$

where

$$
O_{k}=\frac{Q\left(u_{k}\right)-P\left(u_{k}\right)}{\left|Q\left(u_{k}\right)-P\left(u_{k}\right)\right|}
$$

An angular tolerance cone is illustrated in Fig. 3.11 to indicate the limits of tool orientation errors. Similar to the upper b-spline generation, an iterative fitting process is applied until the maximum tool orientation error is below $e_{\theta_{\max }}$.


Figure 3.11: Upper B-spline Generation and Tolerance Cone Definition for Orientation

The algorithm shown in Fig. 3.12 summarizes the least squares fitting of the double bspline toolpath. The resulting knot vector of the upper spline might have more knots then the lower one or the vice versa. To make the trajectories synchronized, both of the knot vectors must be equal. Therefore, abundant knots are inserted into the knot vector of the $b$-splines. The method by Piegl [58] can be used to insert sparse knots into the knot vectors. Added redundant control points do not change the shape of the curve or the parameterization, and the resultant knot vector will preserve desired synchronization.


Figure 3.12: Double B-spline Generation Algorithm

### 3.6. Smoothing the Double Spline Toolpath

### 3.6.1. Introduction

The quality of the fitted curves is crucial for high speed machining. When machining at very high feedrates, the geometrical properties of the curved toolpath directly affects the tracking performance of machine tool. For an improved tracking of the toolpath, the resultant trajectory should deliver very smooth kinematic profiles to the drives.

The dominant factor affecting the tracking performance is identified as the jerk component [28] of the trajectory. The total jerk depends on both the geometric properties of the toolpath such as the change of the curvature, as well as the applied kinematic profiles that build up the complete trajectory. For instance, if the geometric profile shows sharp changes and high roughness, a varying or very conservative feedrate must be applied in the trajectory generation to avoid contouring errors and possible kinematic limit violations of the drives causing possible actuator saturations. Therefore, the geometric toolpath is desired to be as smooth as possible containing minimum changes in the curvature i.e. jerk components. This allows imposing more constant and smooth feed sections in the kinematic profile at the trajectory generation step. As a result, the total machining time will be reduced and the tracking performance of the drives is improved.

Geometric jerk is the third derivative of the position vector, $P(u)$ with respect to the arc length $(s)$. Minimization of a cost function based on geometric jerk can be obtained through nonlinear optimization methods. However, this process introduces a significant computational load to the trajectory generation. In [22] it was proposed that minimizing the third derivative of the position vector with respect to the spline parameter $(u)$ results in significantly reduced jerk profiles as well. Therefore, this approach is adapted in order to minimize the jerk content of the trajectory.

As mentioned, the motivation of this section is to generate a smoother geometric profile of the toolpath, which is easier to track. Two optimization algorithms are presented to modify the double b-spline toolpath. Firstly, the lower b-spline is addressed. Secondly, the orientation of the tool is optimized by modifying the upper spline for smoother rotation of the tool. Machining

### 3.6.2. Smoothing the Lower B-Spline

In the linear least squares fitting process, the quality of the curves mainly depends on the constructed knot vector and the control point positions. The initial knot vector is created by the chord length method that might cause oscillations or twists on the b-spline [56]. Moreover, during the least squares fitting process of the $b$-spline toolpath, extra knots are added into the knot vector (see Fig. 3.12). This can distort the knot distribution and may cause severe fluctuations on the b-spline derivatives. These shortcomings of the least squares approximation are addressed, and the overall smoothness of the toolpath is improved in the following using the jerk minimization objective. The objective in this section is to smooth the lower b-spline of the toolpath by a linear optimization algorithm, which mainly contributes to the motion of cartesian axes of the machine tool.

The tool tip position is given by the lower b-spline as $P(u)=\left[\begin{array}{lll}x^{t}(u) & y^{t}(u) & z^{t}(u)\end{array}\right]^{T}$. The geometric jerk is defined as the third derivative of $P(u)$ with respect to the spline parameter and its minimization can be posed as an integrated square minimization problem as;

$$
\begin{equation*}
J_{K}=\min _{\theta} \int\left(P_{u u u}^{T} \cdot P_{u u u}\right) \mathrm{d} u . \tag{3.36}
\end{equation*}
$$

$\theta_{t}$ is the vector of b-spline control points, $\left(\left[\begin{array}{lll}P_{x} & P_{y} & P_{z}\end{array}\right]^{r}\right)$ and $P_{u u u}$ is the third derivative of the lower b-spline calculates as,
$P_{\text {uuu }}=\left[\begin{array}{l}x_{\text {wuu }}^{t} \\ y_{\text {uuu }}^{t} \\ z_{\text {uuu }}^{t}\end{array}\right]=\frac{\mathrm{d}^{3} P(u)}{\mathrm{d} u^{3}}=\sum_{i=0}^{n} N_{i, P_{\text {muu }}}\left[\begin{array}{l}P_{x} \\ P_{y} \\ P_{x}\end{array}\right]$ where


Expressed in Eq.(3.37), the axis control points $\left(\begin{array}{lll}\left.\left[\begin{array}{lll}P_{x} & P_{y} & P_{z}\end{array}\right]^{r}\right) \text { affect only the third }\end{array}\right.$ derivatives of associated axis components. For instance, modifying the control point $P_{x}$ would change the derivatives of only $x$ axis components. Therefore, $P_{\text {uuu }}$ in each direction can be minimized separately. The jerk objective function $J_{K}$ in Eq. (3.36) is then expressed separately for each axis component as follows:
$J_{K}=\min _{P_{x}, P_{y}, P_{z}} \int\left(P_{u u u}^{T} \cdot P_{u u u}\right) d u \Leftrightarrow\left[\begin{array}{l}J_{K_{x}}=\min _{P_{x}} \int\left(x_{u u u}^{t}\right)^{2} \mathrm{~d} u, \quad J_{K_{y}}=\min _{P_{y}} \int\left(y_{u u u}^{t}\right)^{2} \mathrm{~d} u, \\ \text { and } \quad J_{K_{z}}=\min _{P_{z}} \int\left(z_{u u u}^{t}\right)^{2} \mathrm{~d} u\end{array}\right]$
where $x_{u u u}^{t}, y_{u u u}^{t}, y_{u u u}^{t}$ are the axis derivative components (Eq.(3.37)) of lower b -spline and the $J_{K_{x}}, J_{K_{y}}, J_{K_{z}}$ are the objective functions to be minimized for each axis component, respectively. The solution process of minimizing the jerk objective function $J_{K_{x}}$ for the $x$ component is shown in the following. The integral square of the third derivative (jerk) throughout the b-spline $\left(u \in\left[\begin{array}{ll}0 & 1\end{array}\right]\right)$ is written as,

$$
\begin{equation*}
J_{K_{x}}=\min _{P_{x}} \int_{u=0}^{\mu=1}\left(x_{u u u}^{t} \cdot x_{u u u}^{t}\right) \mathrm{d} u . \tag{3.39}
\end{equation*}
$$

Substituting the expression for $x_{u u u}^{t}$ from Eq. (3.37), the above integral is rewritten as:

$$
\begin{equation*}
J_{K_{x}}=\left(P_{x}\right)^{T}\left(\int_{0}^{1}\left(N_{u u u}^{T} \cdot N_{u u u}\right) \mathrm{d} u\right)\left(P_{x}\right)=\left(P_{x}\right)^{T}(K)\left(P_{x}\right) \tag{3.40}
\end{equation*}
$$

where $N_{u u u}=\left[\begin{array}{llll}N_{0, p \text { uuu }} & (u) & N_{1, p \text { puu }}(u) & N_{2, p u u u}\end{array}(u) \quad \ldots \quad N_{n, p u u u}(u)\right]^{T}$ and $K$ is a $n+1 \times n+1$ symmetric weight matrix, which is only a function of the integral of basis function derivatives. For any given knot distribution, $K$ matrix is constructed for the b -spline parameter interval $u \in\left[\begin{array}{ll}0 & 1\end{array}\right]$ as:

$$
\begin{aligned}
& \text { SYM }
\end{aligned}
$$

When the basis functions are in Bezier form, the weight matrix ( $K$ ) can be expressed analytically since the knot vector is uniform and basis functions are repetitive. Identifying an analytical form for a single basis function will be enough to compute the integrals. On the other hand, during the fitting process of non-uniform $b$-splines, the knot vector changes at each iteration when extra knots are added. It is computationally expensive to calculate each basis function analytically during the least squares fitting. Therefore, the $K$ matrix is computed by numerical integration methods such as using Simpson's $3 / 8$ rule, which is relatively efficient.

The previous optimization problem for the least squares fitting was given by Eq. (3.25) as

$$
\vec{J}_{L_{x}}(P, \Lambda)=\frac{1}{2}\left(q_{x}^{t}-\Phi P_{x}\right)^{T}\left(q_{x}^{\prime}-\Phi P_{x}\right)+\Lambda\left(L \cdot P_{x}-\xi\right)
$$

and it is modified by adding the minimum jerk objective function defined in Eq. (3.40) for the smoothing action. The augmented minimization problem can be represented as:

$$
\begin{align*}
& \min _{P_{x}}\left[J_{L_{x}}+J_{K_{x}}\right]=\min _{P_{x}} \frac{1}{2}[\gamma \underbrace{\left(q_{x}^{\prime}-\Phi P_{x}\right)^{T}\left(q_{x}^{t}-\Phi P_{x}\right)}_{\text {Firing Affect }}+\alpha \underbrace{\left(P_{x}\right)^{T}(K)\left(P_{x}\right)}_{\text {Smootheing Affect }}]\},  \tag{3.42}\\
& \text { Subject to }: L \cdot P_{x}=\xi
\end{align*}
$$

which is also a linear quadratic minimization problem, and can be solved by Introducing the Lagrange multipliers: $\Lambda=\left[\begin{array}{lllll}\Delta_{0} & \Delta_{1} & \Delta_{2} & \cdots & \Delta_{r}\end{array}\right]$, the resultant linear system of equation is obtained as,

$$
\left.\begin{array}{l}
\left(\gamma \Phi^{T} \Phi+\alpha K\right) P_{x}+L^{T} \Lambda=\gamma \Phi^{T} q_{x}^{t}  \tag{3.43}\\
L \cdot P_{x}=\xi
\end{array}\right\} \Rightarrow\left[\begin{array}{cc}
\left(\gamma \Phi^{T} \Phi+\alpha K\right) & L^{T} \\
L & 0
\end{array}\right] \cdot\left[\begin{array}{c}
P_{x} \\
\Lambda
\end{array}\right]=\left[\begin{array}{c}
\gamma \Phi^{T} q_{x}^{t} \\
\xi
\end{array}\right]
$$

where $\alpha$ and $\gamma$ are the weights of the multi-objective minimization problem. $\alpha$ is the weight for smoothing effect and $\gamma$ specifies the fitting effect. Increasing $\gamma$ would increase the fitting affect. Therefore, the overall errors will decrease but the smoothness of the resultant spline would decline. Respectively, $\alpha$ increases the smoothing affect during fitting. The two parameters must be tuned for the fitting requirement. By experience, the parameters can be selected around $\gamma=10^{6} \sim 10^{7}$ and $\alpha=10^{-3} \sim 10^{-2}$. where their affect is investigated during simulations in Section 3.7.

### 3.6.3. Angular Jerk Minimization

In a similar manner to the optimization objective mentioned in the previous section, the angular jerk profile throughout the toolpath can be augmented. Similarly, angular Jerk is defined as the third derivative of the orientation spline $O_{u u u}(u)=\mathrm{d}^{3} O(u) / \mathrm{du}^{3}$. Minimizing the $3^{\text {rd }}$ derivative of the orientation spline helps smoothing the rotational motion of the tool.


Figure 3.13: Tool Axis Vector
The orientation spline was defined in section 3.3 as,

$$
O(u)=\left[\begin{array}{c}
O_{x}  \tag{3.44}\\
O_{y} \\
O_{z}
\end{array}\right]=O_{x} \dot{i}+O_{y} \vec{j}+O_{z} \vec{k}=\frac{A_{x}(u)}{|\vec{A}|} \dot{i}+\frac{A_{y}(u)}{|\vec{A}|} \vec{j}+\frac{A_{z}(u)}{|\vec{A}|} \vec{k}
$$

where the tool axis vector $A=A(u)$ shown in Fig. 3.13 is defined between the upper and lower b-splines of the toolpath as,

$$
A(u)=\left[\begin{array}{l}
A_{x}  \tag{3.45}\\
A_{y} \\
A_{z}
\end{array}\right]=Q(u)-P(u)=\sum_{i=0}^{n}\left(N_{i, p}(u) \cdot\left(\left[\begin{array}{c}
P_{x} \\
P_{y} \\
P_{z}
\end{array}\right]-\left[\begin{array}{l}
Q_{x} \\
Q_{y} \\
Q_{z}
\end{array}\right]\right)\right)
$$

where $\left[\begin{array}{lll}P_{x} & P_{y} & P_{z}\end{array}\right]^{T}$ and $\left[\begin{array}{lll}Q_{x} & Q_{y} & Q_{z}\end{array}\right]^{T}$ are the control points of lower and the upper bsplines, respectively. In order to obtain the derivatives of $O(u)$, the norm of the tool axis vector $|\vec{A}|=\sqrt{A_{x}^{2}+A_{y}^{2}+A_{z}^{2}}$ in Eq. (3.44) and its derivatives with respect to the spline parameter $\left(|A|_{u},|A|_{u u},|A|_{u u u}\right)$ must be computed. The resultant $1^{\text {st }}, 2^{\text {nd }}$ and $3^{\text {rd }}$ derivatives of the orientation spline are expressed as:

$$
\begin{align*}
& \frac{\mathrm{d} O}{\mathrm{~d} u}=\left[\begin{array}{l}
O_{x u} \\
O_{y u} \\
O_{z u}
\end{array}\right]=\frac{A_{u}}{|A|}-\frac{A|A|_{u}}{|A|^{2}},  \tag{3.46}\\
& \frac{\mathrm{~d}^{2} O}{\mathrm{~d} u^{2}}=\left[\begin{array}{l}
O_{x u u} \\
O_{y u u} \\
O_{z u u}
\end{array}\right]=\frac{A_{u}}{|A|}-\frac{A|A|_{u u}-2 A_{u}|A|_{u}}{|A|^{2}}+2 \frac{A|A|_{u}{ }^{2}}{|A|^{3}},  \tag{3.47}\\
& \frac{\mathrm{~d}^{3} O}{\mathrm{~d} u^{3}}=\left[\begin{array}{l}
O_{x u u u} \\
O_{y \text { uuu }} \\
O_{z u u u}
\end{array}\right]=\frac{A_{u u u}|A|-A_{u u}|A|_{u}}{|A|^{2}}-\frac{2|A||A|_{u} A|A|_{u u}+2\left(A_{u}|A|_{u u}+A_{u u}|A|_{u}\right)|A|^{2}-4\left(A_{u}|A|_{u}{ }^{2}|A|\right)}{|A|^{4}} . \\
& -\frac{A_{u}|A|_{u u}+A|A|_{u u u}+\left.2\left(A A_{u}|A|_{u}^{2}+2 A|A|_{u}|A|_{u u}\right) A\right|^{3}-6\left(A|A|_{u}{ }^{3}|A|^{2}\right)}{|A|^{6}} \tag{3.48}
\end{align*}
$$

As shown in Eq. (3.48), the third derivative of the angular spline in each direction depends on $|A|,|A|_{u},|A|_{u u}$ and $|A|_{\text {uuu }}$, which are functions of the upper and lower b-spline axis components i.e. the control points $\left(\left[\begin{array}{lll}P_{x} & P_{y} & P_{z}\end{array}\right]^{T}\right.$ and $\left[\begin{array}{lll}Q_{x} & Q_{y} & Q_{z}\end{array}\right]^{T}$ ). Modifying any control point of the upper b-spline in one direction affects the three axis components ( $O_{x}, O_{y}, O_{z}$ ) of the orientation spline since norm of tool axis vector $(|A|)$ changes. Therefore, the problem is formulated as integral sum of all three derivative components of the orientation spline $O_{\text {хиии }}, O_{\text {уиии }}, O_{\text {zиии }}$ on the unit sphere. The new cost function for the angular jerk minimization process can be written as:

$$
\begin{equation*}
J_{O_{u u}}=\int_{u}\left(O_{x u u u}^{2}+O_{y \text { yuu }}^{2}+O_{z u u u}^{2}\right) \mathrm{d} u . \tag{3.49}
\end{equation*}
$$

After the whole toolpath is fitted; the resultant knot vector is known and fixed. Control points of the upper spline $\left(\left[\begin{array}{lll}Q_{x} & Q_{y} & Q_{z}\end{array}\right]^{\mathrm{T}}\right)$ are the free optimization parameters that can be used to modify the upper b-spline in order to penalize the cost function defined in Eq. (3.49) for the angular jerk. The vector of optimization variables is presented below;

$$
\theta_{h}=\left[\begin{array}{llllllllllllll}
Q_{x_{0}} & Q_{x_{1}} & \cdots & Q_{x_{n}} & \vdots & Q_{y_{0}} & Q_{y_{1}} & \cdots & Q_{y_{n}} & \vdots & Q_{z_{0}} & Q_{z_{1}} & \ldots & Q_{z_{n}} \tag{3.50}
\end{array}\right]^{T} .
$$

The constraints of the optimization problem are:
(1) The max angular error, $e_{\theta_{\max }}$ and
(2) the Euclidian distance between upper and lower b-splines $(|A|)$.

The constraints on orientation error $\left(e_{\theta}\right)$ are imposed in order to respect the machining tolerances and restrict the change in the upper b-spline. The limit on $|A|$ will help to constrain the relative distance between the splines near to $H$, which was pre-defined by the machining process. Since it is difficult to satisfy an equality constraint in nonlinear optimization process, a relaxation band is defined for $|A|$ of $\varepsilon=0.01 \sim 0.25 \mathrm{~mm}$ so that the convergence is obtained faster by imposing inequality constraints instead of equalities. Given for $e_{\theta_{\text {max }}}, H$ and $\varepsilon$, the constraints to be imposed are:

$$
\left.\begin{array}{l}
e_{\theta} \leq e_{\theta_{\max }}  \tag{3.51}\\
H-\varepsilon \leq|A| \leq H+\varepsilon
\end{array}\right\}
$$

[^0]\[

\left.$$
\begin{array}{l}
C_{\theta}\left(u_{k}\right)=e_{\theta}-e_{\theta_{\max }} \leq 0,  \tag{3.52}\\
C_{|A|}^{-}\left(u_{k}\right)=H-\varepsilon-|A| \leq 0, \quad C_{|A|}^{+}\left(u_{k}\right)=|A|-H-\varepsilon \leq 0
\end{array}
$$\right\},
\]

and they are grouped as a compact constraint vector as,

$$
C(u)=\left[\begin{array}{lll}
C_{e_{\theta}} & C_{|A|}^{-} & C_{|A|}^{+} \tag{3.53}
\end{array}\right]^{T} \text { for } 0 \leq u \leq 1 .
$$

The optimization problem is then (Eq. (3.49) and Eq. (3.53))

$$
\begin{equation*}
J_{O_{u u t}}=\min _{\left[Q_{x}\right.} Q_{Q_{y}} \int_{\left.Q_{z}\right]}^{u=1}\left(O_{x=0}^{2}+O_{y u u u u}^{2}+O_{z u u u}^{2}\right) \mathrm{d} u \text { Subject to: } C(u) \leq 0 \text { for } 0 \leq u \leq 1 \tag{3.54}
\end{equation*}
$$

Since the objective function and the constraints have nonlinear relationship with the design variables, the formulation in Eq. (3.54) results in a nonlinear optimization problem unlike in Eq. (3.36). It is solved using MATLAB's Optimization Toolbox. The cost function and its gradients can be calculated analytically when the basis functions are given in analytical form after $b$-spline fitting for the angular optimization. Otherwise, numerical integration methods are used to evaluate the integral in Eq. (3.54) where adaptive Gauss Quadrature or a simpler method such as Simpson's $3 / 8$ can be employed. Constraints are checked only at pre-calculated parameter values ( $u$ ) obtained from Eq. (3.28). The gradient of the constraints can easily be supplied analytically to the algorithm, since they are simply a function of basis functions, $\left.N_{i, p}(u)\right|_{u=u_{k}}$. For the algorithm, the evaluated basis functions are computed once and stored to be called by the algorithm each iteration.

### 3.7. Simulation Results

The feasibility of the toolpath generation algorithm presented in this thesis is shown using simulations. The aim of the first step is to fit the double $b$-spline toolpath to given discrete toolpath points (G01 Commands) with respect to specified tolerance limits. The
machining tolerance applied on aeronautical parts is given around $\sim 0.025$ [ mm ] [1]. The conventional spline fitting algorithms build in CNCs for high speed machining recommend fitting tolerances 0.1 [mm] for rough machining and from 0.05 [ mm ] to 0.01 [ mm ] for finishing operations of free form surfaces [69]. In order to show the potential of the presented fitting algorithm, two fitting tests are presented in the following. The $b$-splines of the generated toolpaths are selected at $5^{\text {th }}$ degree.

The error limit for the first b-spline toolpath generation test is set $0.02[\mathrm{~mm}]$ for tool tip and 0.001 [rad] for the tool axis orientation. The test toolpath is initially discretized by CAM software for five-axis machining using 584 G 01 points. The discrete points of the toolpath and the fitted b-splines can be seen on the Fig. 3.14 (a). Since the error bounds are selected tight, the curve and points are overlapping. The fitting process started with 29 initial control points and required $\sim 2.82$ [ min ] on a P4 2.6 desktop computer running MATLAB to result a b-spline toolpath consisting of 42 control points. The error distributions for the tool - tip positioning and tool axis orientation are given in Fig. 3.15 (a). On straight sections, the fitting error for the tool - tip drops down to 0.002 [ mm ] and less than 0.0001 [rad] for the tool orientation, which indicates that especially on smooth parts, tighter tolerances can be achieved. However, towards the end of the toolpath where there is sharp change in the curvature, the fitting errors are on limits. The resultant maximum positioning error is found as $e_{\max }=0.018941[\mathrm{~mm}]$ for the tool tip and orientation error is calculated as $e_{\theta_{\max }}=5.22 \times 10^{-4}[\mathrm{rad}]=0.029[\mathrm{deg}]$ using Eq. (3.32) and Eq. (3.35).

The toolpath for the second test is shown in Fig. 3.14 (b). It is a round shape with constant change in the curvature that is initially defined with 183 TCP. The b -spline fitting algorithm shown in Fig. 3.15 started with 9 control points and required 25 control points to ensure specified $0.01[\mathrm{~mm}]$ tool tip and $0.0005[\mathrm{rad}]$ tool orientation tolerances. Unlike the first fitting test, the errors are distributed evenly as shown in Fig. 3.15 (b). The maximum tool tip error is computed as $e_{\max }=0.009[\mathrm{~mm}]$ and the max tool orientation error is found as $e_{\theta_{\text {max }}}=4.58 \times 10^{-4}[\mathrm{rad}]=0.026[\mathrm{deg}]$.

The results indicate that very tight tolerances can be achieved by the proposed fitting algorithm in Section 3.5. Especially, for roughing and pre-finishing operations, the method allows generation of $b$-spline toolpaths for high speed five-axis machining.


Figure 3.14: Test Double B-spline Toolpaths Generated from Discrete Points
a) Fitting Errors of Tool-path \#1


Figure 3.15: Fitting Errors of the B-Spline Toolpaths

The effect of smoothing objective presented in Section 3.6.1 is investigated in the following simulations. Firstly, the influence of smoothing coefficients is studied on 45 two dimensional data points shown in Fig. 3.16 for an X-Y table. The fitted b-splines have 20 control points and constrained at both ends to the data points.

In Fig. 3.16 (a) a regular least squares fit without any smoothening affect can be seen. The curve is deteriorated especially at the start section because of the incompetent knot distribution estimated by arc the length approximation. The average error of the fit is $e_{\text {mean }}=0.211[\mathrm{~mm}]$. In Fig. 3.16 (b), the presented jerk minimization objective is implemented into the least squares algorithm using Eq. (3.43). Fitting and smoothening coefficients are selected as $\gamma=1 e 6$ and $\alpha=0.01$ respectively. As a result, he curve shows higher average fitting error of $e_{\text {mean }}=0.404[\mathrm{~mm}]$. But the oscillation at the beginning is eliminated and it displays improved change in the curvature (See Table 3.1). Furthermore, it can be observed by eye-inspection that the optimized curve is smoother. In Fig. 3.16 (c), the fitting coefficient is increased to $\gamma=1 e 7$. As result of this, the average error is decreased to $e_{\text {mean }}=0.255[\mathrm{~mm}]$ with some loss of the curve smoothness. In the last simulation Fig. 3.16 (c), the fitting weight is further increased to $\gamma=1 e 8$, which reduces the average errors down to $e_{\text {mean }}=0.217[\mathrm{~mm}]$. The resultant b -spline is almost identical to the regular least squares fit. Furthermore, in Fig. 3.17, the third derivatives of the fitted curves can be compared to judge the smoothness quality of the curve. Reducing the fitting coefficient diminishes clearly the fluctuations in the $3^{\text {rd }}$ derivative of the spline resulting in a smoother curve. As a result, the acceleration components ( $2^{\text {nd }}$ derivative) on the b-spline (See Fig. 3.17 (a) and Fig. 3.17 (b) ) contains less change. As observed from the simulations, the most suitable coefficients can be selected as $\gamma=1 e 7$ and $\alpha=0.01$ in order to smooth the curve while still preserving an acceptable fitting effect. The comparison of curvature changes is presented in Table 3.1.


Figure 3.16: Multi Objective Least Squares Smoothening of a B-spline - a) Regular Least Squares Fit, b-d) Optimized Least Squares Fit Comparisons

| Fitting \& Smoothing |  |
| :---: | :---: |
| Coefficients |  |
| $((\gamma, \alpha))$ | Integral of Rate of Change of the <br> Curvature |
| $\gamma=1, \alpha=0$ | $\left.\left(\left(\frac{d \kappa}{d s}\right)^{2} d s\right)\right)$ |
| $\gamma=1 e 6, \alpha=0.01$ | $\mathbf{1 . 4 0 9} \mathbf{e}^{17}$ |
| $\gamma=1 e 7, \alpha=0.01$ | $\mathbf{8 . 5 7} \mathbf{e}^{\mathbf{1 5}}$ |
| $\gamma=1 e 8, \alpha=0.01$ | $\mathbf{1 . 2 5 9} \mathbf{e}^{\mathbf{1 6}}$ |
| $\mathbf{1 . 0 4} \mathbf{e}^{\mathbf{1 7}}$ |  |

Table 3.1: Comparison of Curvatures

$$
--\alpha=0 \quad \cdots \cdots=1 \mathrm{e} 8, \alpha=0.01 \quad \ldots \gamma=1 \mathrm{e} 7, \alpha=0.01 \quad-\gamma=1 \mathrm{e} 6, \alpha=0.01
$$



Figure 3.17: $2^{\text {nd }}$ and $3^{\text {rd }}$ Derivative Comparison of Smoothened B-splines

The smoothing effect is applied to the generation of lower b-spline for the first test toolpath shown in Fig. 3.14 (a). The weights for fitting and smoothening effects are chosen as $\gamma=1 e 7$ and $\alpha=0.01$ while setting the maximum error tolerance as $e_{\max }=0.02[\mathrm{~mm}]$. However, using the above fitting and smoothing coefficients, the algorithm did not converge to satisfy the specified tolerance band. Consequently, the weight for smoothing effect is further reduced to $\alpha=0.001$. The smoothness of the $b$-spline fits for the toolpath is measured by comparing the $3^{\text {rd }}$ derivatives of regular and optimized fits shown in Fig. 3.18. As inspected, the quality of fitted b-splines is very close. Because of the tight error limitations and reduced smoothening coefficient, both of the $b$-splines follow almost the same pattern. As a result, tuning of the smoothening effect is quite difficult in case of tight tolerance specifications and larger number of tool center points. For machining operations, the fitted $b$-spline is overly constrained by tolerance requirements allowing less change in the quality. Therefore, the optimization technique should be employed for generation of toolpaths with relaxed tolerance specifications.


Figure 3.18: Comparison of Optimized Lower B-Spline

The angular jerk optimization technique presented in. Section 3.6.2 was applied on the first $b$-spline toolpath shown in Fig. 3.14 (a) to inspect its effectiveness. Optimizing this long toolpath at once proved to be a very slow and inefficient process when considering the large number of control variables and constraints. In order to reduce the total processing time of the non-linear optimization, a technique similar to windowing scheme was implemented. The local modification property [58] of $b$-splines makes it possible that modifying a control point $P_{i}$ affects the curve only in the parameter range of $u \in\left[\begin{array}{ll}\mu_{i} & \mu_{i+p+1}\end{array}\right]$. The upper b-spline is optimized locally at each step without altering shape of the whole curve. Hence, an optimization window size is set as with 5 control points. The window is shifted from the start towards the end of the toolpath while optimizing the section affected only by those control points. In case of non-convergence; the size of the window was increased by including one more control point from the un-optimized region.

The optimization procedure for this toolpath consists of 42 variables described above and $3 \times 584$ constraints to be checked. Using the windowing technique; $\sim 5.1$ [mins] are required on P4 2.6 desktop computer running MATLAB resulting in a feasible offline optimization procedure.

The results for the optimized kinematic profiles of the tool at constant feedrate of 35 [ $\mathrm{mm} / \mathrm{sec}$ ] are presented in Fig. 3.19. The angular acceleration profile presents fluent changes with less fluctuations resulting in a smoother trajectory. Hence, less vibration components are injected into the system while improving the overall tracking performance. The mean angular acceleration and jerk values at constant feed are smoothened and reduced by 9 [\%] and 37 [\%] respectively. The max angular acceleration is reduced by 54 [\%] and the max angular jerk is brought down 79 [\%] after the optimization. The effect of the angular jerk optimization on individual rotary axis profiles is presented in Fig. 3.20 and smoothened linear axis profiles are given in Fig. 3.21. Spikes in the linear axis acceleration and jerk profiles causing possible actuator saturations as well as excessive tracking errors are annihilated. For instance, the highest reduction is observed in the Y axis jerk profile by 80 [\%]. Results indicate that the angular jerk optimization process reduces the overall jerk content of the toolpath efficiently allowing operating the machine tool at higher feedrate without violating the drive's kinematic limits.


Figure 3.19: Angular Kinematic Profiles


Figure 3.20: Rotary Axis Kinematic Profiles


Figure 3.21: Linear Axis Kinematic Profiles

### 3.8. Summary

A continuous smooth representation of the toolpath is significant for higher part quality in high speed five-axis machining. In this chapter, a continuous toolpath consisting of double b -splines is generated from discrete tool path data. The fitting errors are limited depending on the given number of points and toolpath complexity as low in the range of $0.01-0.05[\mathrm{~mm}]$. Therefore, the implementation of such offline b-spline toolpath generation is suitable for most rough, as well as various finishing operations. In addition, smoothing of the toolpath is investigated for part quality and tracking performance by utilizing jerk objectives. The first optimization scheme for the tool tip spline did not found feasible because of tight tolerance specifications applied on the curve. However, the angular jerk of the toolpath is reduced using nonlinear optimization procedure. It is applied after the toolpath is fit and can handle long toolpaths time efficiently. The resultant angular acceleration profile shows smoother trends that enhances the tracking performance of the machine tool, allows increasing the overall feedrate.

## Chapter 4

## Interpolation and Feedrate Profiling for 5-Axis Machining

### 4.1. Introduction

A toolpath in terms of $b$-splines is defined for five-axis machine tools in Chapter 3 of the thesis. The next step is to generate an optimum feedrate profile for five-axis machine tools processing this b -spline toolpath, and interpolate those feed commands to the machine tool axes. The real time interpolation of $b$-spline toolpaths with minimal feed fluctuation is investigated in this Chapter. Furthermore, a jerk continuous and smooth cycle time optimal feedrate profile is generated for five-axis machining.

Feedrate fluctuations arising from incorrect spline parameter calculation is discussed in the Section 4.2, and the 'Feed Correction Polynomial' method, developed by Erkorkmaz and Altintas [22] is implemented on $b$-spline toolpaths for feed consistent interpolation. Results from the 'Feedrate Correction Polynomial' approach are compared to the widely used, 'Taylor Series Expansion' [50] method in the simulations section. The Taylor approximation is also briefly explained.

A feedrate scheduling algorithm was proposed by the same authors processing quintic spline toolpaths on Cartesian machines [3]. In the feedrate profiling work, the algorithm is modified for the double B-Spline toolpaths and implemented considering the 5-Axis machine tool's kinematic transformation and drive limits. The algorithm generates smooth kinematic profiles that consider machine tool's kinematic limits such as velocity, acceleration, jerk for minimum cycle time while aiming for better tracking performance.

### 4.2. Interpolation with Minimal Feed Fluctuation

For a specified feedrate profile $V=V(t)$, a resulting displacement profile $s=s(t)$ is obtained. Then, the position commands to the drives are calculated in order to track the
specified displacement profile during the machining operation. This task is performed by the real time feedrate interpolator embedded within the machine tool controller. The displacement profile can be provided in a linear, circular or a spline toolpath [42] format. For any of these provided toolpaths, the feedrate interpolator must calculate the axis increments as reference commands to the control loop at each sampling interval $\left(T_{s}\right)$. This task becomes computationally expensive when processing spline toolpaths [74].

In Fig. 4.1, a parametric b-spline toolpath is illustrated with a given displacement profile. As the spline parameter is incremented $(\Delta u)$ this corresponds to an arc increment $(\Delta s)$ on the b-spline. The travel distances (i.e., the arc increments, $(\Delta s)$ ) at sampling intervals are scheduled during the feedrate generation step. Hence, the task of the real-time interpolator is basically to increment the spline parameter $u=u(t)$ accurately to satisfy the desired arc increment $s=s(t)$.


Figure 4.1: Real-time Spline Interpolation

In particular, b -spline toolpaths are defined in the parameter interval of $u \in\left[\begin{array}{ll}0 & 1\end{array}\right]$. A straightforward approach for real-time interpolators is to linearly increment the b -spline parameter proportional to the ratio of the arc increment to the total arc length $\left(S_{\text {total }}\right)$ (i.e.,
$\left.\Delta u=1 / S_{\text {total }} \cdot \Delta s\right)$. However, this linear relationship between spline parameter $(u)$ and arc displacement $(s)$ is not valid over the entire b -spline [21]. An imperfect calculation of the spline parameter would cause fluctuations on the actual feedrate. Such fluctuations would result in higher accelerations and jerk components on the drives, causing unwanted vibrations of the tool and deterioration from the part quality.

### 4.2.1. Taylor Series Approximation

The Taylor series approximation for real time spline interpolation was first introduced by Shipitalni et. al. [67] to estimate the next point along a spline toolpath on constant feedrate. In the literature, this method is widely used for interpolation of the b-spline toolpaths [74, 87, 92, 93]. It is based on calculating the curve parameter by using the differential relationship between the arc length $(s)$ and spline parameter $(u)$ when the sampling interval is sufficiently small.

In the practice of five-axis machining, the feedrate is practically defined on the tool tip, along the lower b -spline (tool tip spline) of the toolpath $P(u)$, the feedrate can be related to the spline parameter $(u)$ using differential geometry as follows:

$$
\begin{equation*}
V(t)=\frac{\mathrm{d} s}{\mathrm{~d} t}=\frac{\mathrm{d} s}{\mathrm{~d} u} \frac{\mathrm{~d} u}{\mathrm{~d} t} \Rightarrow \frac{\mathrm{~d} u}{\mathrm{~d} t}=\frac{f(t)}{\mathrm{d} s / \mathrm{d} u} \tag{4.1}
\end{equation*}
$$

The differential relationship between the arc length and spline parameter on $P(u)=\left[\begin{array}{lll}x^{t}(u) & y^{t}(u) & z^{t}(u)\end{array}\right]^{\mathrm{T}}$ can be expressed as:

$$
\left.\begin{array}{l}
\mathrm{d} s=\left|\frac{\mathrm{d} P(u)}{\mathrm{d} u}\right| \cdot \mathrm{d} u=\left|P_{u}(u)\right| \cdot \mathrm{d} u=\sqrt{\left(x_{u}^{t}(u)\right)^{2}+\left(y_{u}^{t}(u)\right)^{2}+\left(z_{u}^{t}(u)\right)^{2}} \mathrm{~d} u  \tag{4.2}\\
\frac{\mathrm{~d} u}{\mathrm{~d} s}=\frac{1}{\left|P_{u}(u)\right|}=\frac{1}{\sqrt{\left(x_{u}^{t}(u)\right)^{2}+\left(y_{u}^{t}(u)\right)^{2}+\left(z_{u}^{t}(u)\right)^{2}}}
\end{array}\right\} .
$$

The subscript variables are used to denote differentiation with respect to that variable as,

$$
\left.\begin{array}{l}
x_{u}^{t}=\mathrm{d} x^{t} / \mathrm{du}=\sum_{i=0}^{n} N_{i, p}^{(1)}(u) \cdot P_{x i}=\sum_{i=0}^{n}\left(\frac{\mathrm{~d} N_{i, p}(u)}{\mathrm{du}} \cdot P_{x i}\right) \\
y_{u}^{t}=\mathrm{d} y^{t} / \mathrm{du}=\sum_{i=0}^{n} N_{i, p}^{(1)}(u) \cdot P_{y i}=\sum_{i=0}^{n}\left(\frac{\mathrm{~d} N_{i, p}(u)}{\mathrm{du}} \cdot P_{y i}\right)  \tag{4.3}\\
z_{u}^{t}=\mathrm{d} z^{t} / \mathrm{du}=\sum_{i=0}^{n} N_{i, p}^{(1)}(u) \cdot P_{z i}=\sum_{i=0}^{n}\left(\frac{\mathrm{~d} N_{i, p}(u)}{\mathrm{du}} \cdot P_{z i}\right)
\end{array}\right\}
$$

and similarly,

$$
\left.\begin{array}{l}
x_{u u}^{t}=\mathrm{d}^{2} x^{t} / \mathrm{du}^{2}=\sum_{i=0}^{n} N_{i, p}^{(2)}(u) \cdot P_{x i}=\sum_{i=0}^{n}\left(\frac{\mathrm{~d}^{2} N_{i, p}(u)}{\mathrm{du}^{2}} \cdot P_{x i}\right) \\
y_{u u}^{t}=\mathrm{d}^{2} y^{t} / \mathrm{du}^{2}=\sum_{i=0}^{n} N_{i, p}^{(2)}(u) \cdot P_{y i}=\sum_{i=0}^{n}\left(\frac{\mathrm{~d}^{2} N_{i, p}(u)}{\mathrm{du} \mathrm{u}^{2}} \cdot P_{y i}\right)  \tag{4.4}\\
z_{u u}^{t}=\mathrm{d}^{2} z^{t} / \mathrm{du}^{2}=\sum_{i=0}^{n} N_{i, p}^{(2)}(u) \cdot P_{z i}=\sum_{i=0}^{n}\left(\frac{\mathrm{~d}^{2} N_{i, p}(u)}{\mathrm{du}^{2}} \cdot P_{z i}\right)
\end{array}\right\}
$$

where $N_{i, p}(u)$ is the $\mathrm{p}^{\text {th }}$ degree basis functions, $N_{i, p}^{(k)}(u)$ is the $\mathrm{k}^{\text {th }}$ derivative computed as expressed in Chapter 3 by,

$$
\begin{equation*}
N_{i, p}(u)=\frac{u-\mu_{i}}{\mu_{i+1}-\mu_{i}} N_{i, p-1}(u)+\frac{\mu_{i+p+1}-u}{\mu_{i+p+1}-\mu_{i+1}} N_{i+1, p-1} \tag{4.5}
\end{equation*}
$$

and $\frac{\mathrm{d}^{(k)} N_{i, p}(u)}{\mathrm{d} u^{(k)}}=p\left(\frac{\mathrm{~d}^{(k-1)} N_{i, p-1}(u) / \mathrm{d} u^{(k-1)}}{\mu_{i+p}-\mu_{i}}-\frac{\mathrm{d}^{(k-1)} N_{i+1, p-1}(u) / \mathrm{d} u^{(k-1)}}{\mu_{i+p+1}-\mu_{i+1}}\right)$.

Substituting Eq. (4.2) into Eq. (4.1) yields,

$$
\begin{equation*}
\frac{\mathrm{d} u}{\mathrm{~d} t}=\frac{V(t)}{\sqrt{\left(x_{u}^{t}(u)\right)^{2}+\left(y_{u}^{t}(u)\right)^{2}+\left(z_{u}^{t}(u)\right)^{2}}} \tag{4.7}
\end{equation*}
$$

A computationally efficient solution for Eq. (4.7) is necessary in order to achieve a realtime interpolation of the parametric curve; an approximation based on Taylor's expansion is used. For instance, when the curve has a large radius of curvature and the sampling interval is sufficiently small enough (e.g., $T_{s} \cong 1 \mathrm{msec}$ ) a first order approximation is usually adequate at constant feedrate;

$$
\begin{equation*}
u_{i+1}=u_{i}+\left.T_{s} \frac{\mathrm{~d} u}{\mathrm{~d} t}\right|_{u=u_{i}}=u_{i}+\frac{V \cdot T_{s}}{\sqrt{\left(x_{u}^{t}\left(u_{i}\right)\right)^{2}+\left(y_{u}^{t}\left(u_{i}\right)\right)^{2}+\left(z_{u}^{t}\left(u_{i}\right)\right)^{2}}} . \tag{4.8}
\end{equation*}
$$

However, when the curve has small radii of curvature, the second order approximation is necessary, which is expressed as,

$$
\begin{equation*}
u_{i+1}=u_{i}+\left.T_{s} \frac{\mathrm{~d} u}{\mathrm{~d} t}\right|_{u=u_{i}}+\left.\frac{T_{s}^{2}}{2} \frac{\mathrm{~d}^{2} u}{\mathrm{~d} t^{2}}\right|_{u=u_{i}} \tag{4.9}
\end{equation*}
$$

where $\left(\mathrm{d}^{2} u / \mathrm{d} t^{2}\right)$ is calculated by taking the derivative of Eq. (4.2):

$$
\left.\begin{array}{l}
\frac{\mathrm{d}^{2} u}{\mathrm{~d} t^{2}}=\frac{\mathrm{d}(\mathrm{~d} u / \mathrm{d} t}{\mathrm{d} u} \frac{\mathrm{~d} u}{\mathrm{~d} t}, \\
\quad=-V(t) \frac{x_{u}^{t}(u) x_{u u}^{t}(u)+y_{u}^{t}(u) y_{u u}^{t}(u)+z_{u}^{t}(u) z_{u u}^{t}(u)}{\left(\left(x_{u}^{t}(u)\right)^{2}+\left(y_{u}^{t}(u)\right)^{2}+\left(z_{u}^{t}(u)\right)^{2}\right)^{3 / 2}} \cdot \frac{V(t)}{\sqrt{\left(x_{u}^{t}(u)\right)^{2}+\left(y_{u}^{t}(u)\right)^{2}+\left(z_{u}^{t}(u)\right)^{2}}}
\end{array}\right\}
$$

Substituting Eq.(4.10) into Eq.(4.9) yields the second order Taylor series approximation:

$$
\begin{align*}
& u_{i+1}=u_{i}+\frac{V \cdot T_{s}}{\sqrt{\left(x_{u}^{t}\left(u_{i}\right)\right)^{2}+\left(y_{u}^{t}\left(u_{i}\right)\right)^{2}+\left(z_{u}^{t}\left(u_{i}\right)\right)^{2}}}-  \tag{4.11}\\
& \qquad \frac{\left(V \cdot T_{s}\right)^{2}}{2} \frac{x_{u}^{t}\left(u_{i}\right) x_{u u}^{t}\left(u_{i}\right)+y_{u}^{t}\left(u_{i}\right) y_{u u}^{t}\left(u_{i}\right)+z_{u}^{t}\left(u_{i}\right) z_{u u}^{t}\left(u_{i}\right)}{\left(x_{u}^{t}\left(u_{i}\right)\right)^{2}+\left(y_{u}^{t}\left(u_{i}\right)\right)^{2}+\left(z_{u}^{t}\left(u_{i}\right)\right)^{2}}
\end{align*}
$$

At each time step, the spline parameter is calculated by the first order approximation using Eq.(4.8), or for the second order approximation by Eq.(4.11). Following that, the calculated spline parameter is simply used in order to compute the position and orientation values on the b-spline toolpath for 5-Axis machining as mentioned in Chapter 3 of the thesis.

However, the Taylor series expansion is not reliable when the feedrate is varying and results in an imperfect curve parameter calculation for toolpaths having sharp changes in the curvature. Furthermore, its real time implementation is quite computationally expensive for b-spline toolpaths [12] because of the calculation of basis functions and their derivatives. For instance, the basis functions and their $\mathrm{k}^{\text {th }}$ derivatives are calculated recursively by Eq.(4.5) and Eq.(4.6).


Figure 4.2: Basis Function Computation a) Recursive Calculation of Basis Functions --- b) Recursive Calculation of Basis Function Derivatives

As illustrated in Fig. 4.2, the recursive computation of the basis function $N_{i, p}(u)$ requires $\left(2^{p}-1\right)$ function calls to Eq.(4.5) [12]. In addition, regarding to the recursive computation scheme, the calculation the $\mathrm{k}^{\text {th }}$ derivative of a basis function $N_{i, p}^{k}(u)$ requires $\left(2^{k}-1\right)$ function calls to Eq.(4.6) and $2^{k}\left(2^{p-k}-1\right)$ function calls to Eq.(4.5). Hence, for the $\mathrm{k}^{\text {th }}$ derivative of a basis function, total number of $\left(2^{p}-1\right)$ function calls to Eq. (4.5) and Eq. (4.6) are required. Using Eq.(4.3), for a b-spline consisting of $n+1$ control points, the
computation of a derivative at one point on the b-spline demands $(n+1)\left(2^{p}-1\right)$ function call to Eq. (4.5) and Eq. (4.6). The computational load is quite significant when the degree of the b-spline and number of axis are increased. For example, the computation of a point on a cubic b-spline ( $p=3$ ) with 5 control points ( $n=6$ ), 49 number of function calls are required where a quintic b-spline with the same number of control points demands 217 function calls.

### 4.2.2. Feed Correction Polynomial Method

Due to the deficiencies imposed by the Taylor's approximation, the feed correction polynomial method is introduced by Erkorkmaz and Altintas [22] for real time spline interpolation. In this subsection, the feed correction polynomial interpolation method is explained and applied specifically to the b-spline toolpaths.

The feedrate is defined on the lower spline of the toolpath, which is a $p^{\text {th }}$ degree $b$-spline parameterized by a knot vector $(U)$ consisting of $n+p+2$ knots $\left(\mu_{i}\right)$. Subtracting the $p+1$ repetitive knots at the beginning and the end of the knot vector, there are in total $n-p$ internal knots in $U$. This yields to $N_{s}=n-p+1$ internal knot spans in the b-spline representation. The knot spans of the b -spline and their parameter ranges are given as

$$
U=[0 \ldots \overbrace{0, \quad \mu_{\underbrace{}_{p}}^{k=0} ; \mu_{p+1}}^{k=\ldots, \mu_{n-1}}, \overbrace{\mu_{n}, \cdots 1 \ldots 1}^{k=n-p}]
$$

and illustrated in following Fig. 4.3.


Figure 4.3: B-spline Knot Spans and Segmentation

Since the parameter ranges are fixed by the consecutive knots, each internal knot span can be assessed as a piecewise spline segment. Therefore, the whole $b$-spline representation is divided into $n-p+1$ curve segments (See Fig. 4.3). As given in Eq.(4.12), the parameter range for the $\mathrm{k}^{\text {th }}$ segment of the b -spline is presented as:

$$
\begin{align*}
& \mu_{k+p} \leq u \leq \mu_{k+p+1} \\
& \text { where }  \tag{4.12}\\
& k=0, \ldots, n-p
\end{align*}
$$

The arc length, $S_{k}$ for $\mathrm{k}^{\text {th }}$ segment is calculated by dividing each knot span into $\left(M_{k} \geq 100\right)$ subdivisions proportional to its parameter range and computing the corresponding axis positions as:

$$
\left.\begin{array}{l}
x_{k, \ell}^{t}=\sum_{i=0}^{n} N_{i, p}\left(\mu_{k+p}+\ell \cdot \Delta u_{k}\right) \cdot P_{x i} \\
y_{k, \ell}^{t}=\sum_{i=0}^{n} N_{i, p}\left(\mu_{k+p}+\ell \cdot \Delta u_{k}\right) \cdot P_{y i}  \tag{4.13}\\
z_{k, \ell}^{t}=\sum_{i=0}^{n} N_{i, p}\left(\mu_{k+p}+\ell \cdot \Delta u_{k}\right) \cdot P_{z i}
\end{array}\right\}, \begin{gathered}
\\
\Delta u_{k}=\frac{\mu_{k+p+1}-\mu_{k+p}}{M_{k}} \\
\ell=0,1,2, \ldots M_{k}
\end{gathered} .
$$

The segment lengths are computed by summing up the arc increments as:

$$
\left.\begin{array}{l}
S_{k}=\sum_{\ell=1}^{M_{k}} \sqrt{\left(x_{k, \ell}^{t}-x_{k, \ell-1}^{t}\right)^{2}+\left(y_{k, \ell}^{t}-y_{k, \ell-1}^{t}\right)^{2}+\left(z_{k, \ell}^{t}-z_{k, \ell-1}^{t}\right)^{2}}  \tag{4.14}\\
S_{\text {total }}=\sum_{k=0}^{n-p} S_{k}
\end{array}\right\} .
$$

The non-linear relationship between the b-spline parameter $(u)$ and the arc increment ( $\Delta s$ ) was already expressed in Eq.(4.2). According to the feed correction polynomial method, this non-linear relationship can be expressed by another polynomial. The objective is to use the feed correction polynomial in order to calculate the spline parameter as accurately as possible and with less computational load during real time interpolation of $b$-splines. Following that, the computed spline parameter is used to extract the orientation and position values on the double $b$-spline toolpath.

b) Fitted Polynomials Over Knot Spans


Figure 4.4: Feed Correction Polynomial for B-spline Toolpath

The feed correction polynomial is illustrated in Fig. 4.4 (a). For a b-spline toolpath, it is specifically defined between the arc increments ( $\Delta s$ ) and corresponding parameter changes ( $\Delta u$ ) of individual knot spans (segments) as shown in Fig. 4.4 (b). The feed correction polynomial segments in Fig. 4.4 (b) are connected to each other by continuity conditions in order to retain a single polynomial for the whole $b$-spline.

Using seventh order polynomial spline [19], the feed correction polynomial for the $\mathrm{k}^{\text {th }}$ segment of the $b$-spline is written as:

$$
\left.\begin{array}{l}
\hat{u}=A_{k}^{f} s^{7}+B_{k}^{f} s^{6}+\ldots H_{k}^{f}  \tag{4.15}\\
\text { where } \\
\mu_{k+p} \leq \hat{u} \leq \mu_{k+p+1} \text { and } \\
0 \leq s \leq S_{k}
\end{array}\right\} .
$$

The feed polynomial defined in Eq.(4.15) basically supplies an accurate estimate of the b-spline parameter $\hat{u}$ to achieve the desired arc increment within each knot span during the interpolation. The parameter range $(s)$ is bounded by the arc length of knot span $\left(S_{k}\right)$. Considering Eq.(4.13) and Eq.(4.14) for the $\mathrm{k}^{\text {th }}$ segment, the parameter increment and associated arc travels are obtained offline, while computing the arc lengths as:

$$
u=\left[\begin{array}{c}
\mu_{k+p} \\
\mu_{k+p}+\Delta u_{k} \\
\mu_{k+p}+2 \Delta u_{k} \\
\vdots \\
\mu_{k+p+1}
\end{array}\right] \Rightarrow s=\left[\begin{array}{c}
0 \\
s_{1} \\
s_{2} \\
\vdots \\
s_{M_{k}}
\end{array}\right],
$$

$$
\begin{align*}
& \text { where }  \tag{4.16}\\
& \Delta s_{k, \ell}=\sqrt{\left(x_{k, \ell}^{t}-x_{k, \ell-1}^{t}\right)^{2}+\left(y_{k, \ell}^{t}-y_{k, \ell-1}^{t}\right)^{2}+\left(z_{k, \ell}^{t}-z_{k, \ell-1}^{t}\right)^{2}} \\
& s_{\ell}=\sum_{\ell=1}^{M_{k}} \Delta s_{k, \ell},
\end{align*}
$$

In order to avoid any ill conditioning, the parameter ranges in the feed correction polynomial are normalized by the arc length of each knot span $\left(S_{k}\right)$,

$$
\left.\begin{array}{l}
\hat{u}=a_{k}^{f} \sigma^{7}+b_{k}^{f} \sigma^{6}+\ldots h_{k}^{f}  \tag{4.17}\\
\mu_{k+p} \leq \hat{u} \leq \mu_{k+p+1} \\
\text { and } \\
0 \leq \sigma \leq 1
\end{array}\right\} .
$$

where $\sigma=s / S_{k}$ and that results the upper bound of the arc displacement as 1 . The normalized polynomial coefficients are written as $a_{k}^{f}=S_{k}^{7} A_{k}^{f}, b_{k}^{f}=S_{k}^{6} B_{k}^{f}, \ldots, h_{k}^{f}=H_{k}^{f}$. The
first and second derivatives of the spline parameter with respect to the arc displacement $\left(\hat{u}_{s}=d \hat{u} / d s\right.$ and $\left.\hat{u}_{s}=d^{2} \hat{u} / d s^{2}\right)$ are also expressed in normalized formulation:

$$
\left.\begin{array}{l}
\hat{u}_{s}=\frac{d \hat{u}}{d s}=\frac{1}{S_{k}}\left[7 a_{k}^{f} \sigma^{6}+6 b_{k}^{f} \sigma^{s}+\ldots+g_{k}^{f}\right]  \tag{4.18}\\
\hat{u}_{s s}=\frac{d^{2} \hat{u}}{d s^{2}}=\frac{1}{S_{k}^{2}}\left[42 a_{k}^{f} \sigma^{6}+30 b_{k}^{f} \sigma^{5}+\ldots+2 f_{k}^{f}\right]
\end{array}\right\} .
$$

The feed correction polynomial is simply obtained by applying a least squares fit over the normalized arc displacements and the spline parameter values (Eq.(4.16)) at each knot span as:

$$
\hat{u}_{k}=\left[\begin{array}{c}
\mu_{k+p}  \tag{4.19}\\
\mu_{k+p+1}+\Delta u_{k} \\
\mu_{k+p+1}+2 \Delta u_{k} \\
\vdots \\
\mu_{k+p+1}
\end{array}\right]=\underbrace{\left[\begin{array}{ccccc}
0 & 0 & 0 & \ldots & 1 \\
\sigma_{1}^{7} & \sigma_{1}^{6} & \sigma_{1}^{5} & \ldots & 1 \\
\sigma_{2}^{7} & \sigma_{2}^{6} & \sigma_{2}^{5} & \ldots & 1 \\
\vdots & \vdots & \vdots & \ddots & 1 \\
1 & 1 & 1 & \ldots & 1
\end{array}\right]}_{\Phi} \underbrace{\left[\begin{array}{c}
a_{k}^{f} \\
b_{k}^{f} \\
c_{k}^{f} \\
\vdots \\
h_{k}^{f}
\end{array}\right]}_{\theta_{f}} .
$$

Referring to the least squares method, the objective function to be minimized is:

$$
\begin{equation*}
J=\frac{1}{2} e^{T} e=\frac{1}{2}\left(u-\Phi \theta_{f}\right)^{T}\left(u-\Phi \theta_{f}\right) . \tag{4.20}
\end{equation*}
$$

The first and second derivatives of the spline parameter with respect to the arc length can be written as:

$$
\begin{equation*}
u_{s}=\frac{d u}{d s}=\frac{1}{\sqrt{\left(x_{u}^{t}\right)^{2}+\left(y_{u}^{t}\right)^{2}+\left(z_{u}^{t}\right)^{2}}}, \quad u_{s s}=\frac{d^{2} u}{d s^{2}}=\frac{x_{u}^{t} x_{u u}^{t}+y_{u}^{t} y_{u u}^{t}+z_{u}^{t} z_{u u}^{t}}{\left(x_{u}^{t}\right)^{2}+\left(y_{u}^{t}\right)^{2}+\left(z_{u}^{t}\right)^{2}} . \tag{4.21}
\end{equation*}
$$

Therefore, the first and second order constraints at the beginning and the end of the $\mathrm{k}^{\text {th }}$ knot span are given as:

The continuity boundary constraints have to be matched by the feed correction polynomial in order to connect individual polynomials fitted for knot spans over the $b$-spline toolpath. Accordingly, the ranges for $b$-spline parameter and arc travels are defined as zero boundary conditions on the polynomial. Those zero boundary conditions for $\mathrm{k}^{\text {th }} \mathrm{knot}$ span are given as:

$$
\left.\begin{array}{l}
\sigma=0 \Leftrightarrow u=\mu_{k+p}  \tag{4.23}\\
\sigma=1 \Leftrightarrow u=\mu_{k+p+1}
\end{array}\right\} .
$$

Using Eq.(4.17) and Eq.(4.23), the zero, first and second boundary conditions are written in matrix form as:

$$
\left[\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1  \tag{4.24}\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
42 & 30 & 20 & 12 & 6 & 2 & 0 & 0
\end{array}\right]\left[\begin{array}{c}
a_{k}^{f} \\
b_{k}^{f} \\
c_{k}^{f} \\
d_{k}^{f} \\
e_{k}^{f} \\
f_{k}^{f} \\
g_{k}^{f} \\
h_{k}^{f}
\end{array}\right]=\left[\begin{array}{c}
\mu_{k+p} \\
S_{k} u_{s}^{\text {init }} \\
S_{k}^{2} u_{s s}^{\text {init }} \\
\mu_{k+p+1} \\
S_{k} u_{s}^{\text {final }} \\
S_{k}^{2} u_{s s}^{\text {final }}
\end{array}\right] .
$$

Hence, the feed correction polynomial is obtained by solving the linear quadratic optimization problem similar to the one in the Chapter 3 by introducing the vector of Lagrange multipliers, $\Lambda=\left[\begin{array}{llll}\Lambda_{1} & \Lambda_{2} & \ldots & \Lambda_{6}\end{array}\right]^{\mathrm{T}}$. The resultant linear equation system is:

$$
\left.\begin{array}{l}
\Phi^{T} \Phi \theta_{f}+L^{T} \cdot \Lambda=\Phi^{T} u  \tag{4.25}\\
L \theta_{f}=\xi
\end{array}\right\} \Rightarrow\left[\begin{array}{cc}
\Phi^{T} \Phi & L^{T} \\
L & 0
\end{array}\right]\left[\begin{array}{c}
\theta_{f} \\
\Lambda
\end{array}\right]=\left[\begin{array}{c}
\Phi^{T} u \\
\xi
\end{array}\right] .
$$

Solution of Eq. (4.25) gives the coefficients $\theta_{f}$ of the feed correction polynomial. Following the solution, the coefficients are de-normalized as; $A_{k}^{f}=a_{k}^{f} / S_{k}^{7}, \quad B_{k}^{f}=b_{k}^{f} / S_{k}^{6}, \ldots, H_{k}^{f}=h_{k}^{f}$.

### 4.2.3. Iterative Solution of the Spline Parameter

The feed correction polynomial is defined by Eq.(4.17) and the coefficients of the polynomial $\left(A_{k}^{f}, B_{k}^{f}, \ldots, H_{k}^{f}\right)$ are obtained from Eq.(4.25). However, when the feed correction polynomial is directly used in the real time interpolation; it introduces imperfection into the $b$-spline parameter calculation because of the residual errors in the least squares fit. Therefore, the feed correction polynomial interpolation method can be combined with an iterative method in order to solve for the exact $b$-spline parameter during real-time interpolation as shown in Fig. 4.5 (c).
a) Taylor's Approximation

$$
s(t) \sqrt{u_{i+1}=u_{i}+\frac{V(t) T_{s}}{x^{2}\left(u_{i}\right)+y^{2}\left(u_{i}\right)+z^{2}\left(u_{i}\right)}} u(t) \quad \begin{aligned}
& \text { B-Spline } \\
& \text { Toolpath }
\end{aligned}
$$

b) Feed Correction Interpolation

c) Feed Correction with Raphson Iteration


Figure 4.5: Real Time Interpolation Methods

From the feed correction polynomial, the initial guess $u_{\text {init }}=\hat{u}$ for the $b$-spline parameter is obtained. Following that, the desired arc displacement $\Delta s_{d e s}$ for the control interval is calculated using backward differencing of the displacement profile $s(t)$. The arc increment $\Delta s_{\text {test }}$ corresponding to the tested spline parameter $u_{\text {test }}$ is estimated by computing the resulting axis increments:

$$
\Delta s_{\text {test }}=\sqrt{\left(\Delta t_{\text {test }}^{t}\right)^{2}+\left(\Delta \Delta y_{\text {test }}^{t}\right)^{2}+\left(\Delta z_{\text {test }}^{t}\right)^{2}}
$$

where:

$$
\begin{align*}
& \Delta x_{\text {test }}^{t}=x_{\text {test }}^{t}-x_{\text {prev }}^{t}, \Delta y_{\text {test }}^{t}=y_{\text {test }}^{t}-y_{\text {prev }}^{t} \text { and } \Delta z_{\text {test }}^{t}=z_{\text {test }}^{t}-z_{\text {prev }}^{t} \\
& x_{\text {test }}^{t}=\sum_{i=0}^{n} N_{i, p}\left(u_{\text {test }}\right) P_{x i},  \tag{4.26}\\
& y_{\text {test }}^{t}=\sum_{i=0}^{n} N_{i, p}\left(u_{\text {test }}\right) P_{y i} \\
& z_{\text {test }}^{t}=\sum_{i=0}^{n} N_{i, p}\left(u_{\text {test }}\right) P_{z i}
\end{align*}
$$

where $x_{\text {prev }}^{t}, y_{\text {prev }}^{t}$ and $z_{\text {prev }}^{t}$ are the axis commands applied in the previous control sample. The error between the desired and tested arc increments is:

$$
\begin{equation*}
e=\Delta s_{d e s}-\Delta s_{\text {test }}, \tag{4.27}
\end{equation*}
$$

and its gradient with respect to the tested spline parameter is derived using Eq. (4.26) and Eq. (4.27),

$$
\begin{equation*}
\frac{\mathrm{d} e}{\mathrm{~d} u_{\text {test }}}=\frac{\Delta x_{\text {test }}^{t} \cdot\left(d x_{\text {test }}^{t} / d u\right)+\Delta y_{\text {test }}^{t} \cdot\left(d y_{\text {test }}^{t} / d u\right)+\Delta z_{\text {test }}^{t} \cdot\left(d z_{\text {test }}^{t} / d u\right)}{\Delta s_{\text {test }}} \tag{4.28}
\end{equation*}
$$

where $\left(d x_{\text {test }}^{\prime} / d u\right),\left(d y_{\text {test }}^{\prime} / d u\right)$ and $\left(d z_{\text {test }}^{\prime} / d u\right)$ are simply obtained by substituting the calculated value of $u_{\text {test }}$ into the derivative expressions given in Eq.(4.3) and (4.4). The correct value of the spline parameter is computed regards the convergence of the below iteration $(i)$ :

$$
\begin{equation*}
\left(u_{\text {test }}\right)_{i+1}=\left(u_{\text {test }}\right)_{i}-\frac{e_{i}}{\left(\mathrm{de} / \mathrm{d} u_{\text {test }}\right)_{i}} . \tag{4.29}
\end{equation*}
$$

The desired convergence is achieved when the arc increment error $e$ is below $10^{-8}$ as in Eq. (3.28). The fact that a very close guess ( $\hat{u}$ ) is given by the feed correction polynomial and the analytical gradient ( $\mathrm{d} e / \mathrm{d} u_{\text {test }}$ ), the above iteration converges within 2 to 3 iterations, which is convenient for real time interpolation of b-spline toolpaths.

### 4.3. Feedrate Profiling for 5-Axis Machine Tools

The feedrate profile of the trajectory is crucial for the quality of the machining process. In high speed machining, discontinuities and high frequency harmonics of the reference trajectory can easily excite the natural modes of the structure and servo system causing degradation from axis tracking and contouring performance. Therefore, the tangential feed profile defined together with the spline toolpath must ensure high continuity and smoothness allowing the axis servo drives to track the input reference kinematic profiles better.

For basic linear and circular toolpaths, the feedrate profile consists of 3 main sections: acceleration, constant velocity (cruise) and deceleration. The starting acceleration and ending deceleration segments are defined with high smoothness, and their maximum values are determined with respect to individual servo drives acceleration and jerk limits. The cruise velocity is selected as high possible determined by the process parameters and axis kinematic limits aiming for the minimum machining time. On the other hand, b -spline toolpaths enable continuous change in the direction of the tool tip velocity and the tool axis orientation. Even interpolation at constant feedrate can produce excessive acceleration and jerk components on the drives when there is a sharp change in the tool orientation or in the spline curvature.

Therefore, the feed must be scheduled optimally in a smooth way throughout the spline toolpath to avoid violating any axis kinematic limits. For instance, the tangential feedrate at the tool tip must be reduced at sections where the change in the curvature is sudden to prevent exceeding acceleration limits of the drives and increased at straight sections to reduce the overall machining time. In addition, the non-linear inverse kinematics transformation from the workpiece coordinates to the axis commands must be considered for the five-axis machine tools.


Figure 4.6: Overview of Feedrate Profiling

In order to generate a continuously varying feedrate profile, Altintas and Erkorkmaz [3] defined the arc displacement profile as a jerk continuous ( $\mathrm{C}^{3}$ ) fifth degree spline over the specified travel distances on the spline toolpath, and optimized the feed profile for minimum cycle time. They also employed the minimum jerk criteria in the formulation for a smoother displacement profile. Similarly, a quintic displacement profile can be generated as a function of the time durations required to travel each knot span for the $b$-spline toolpaths. The overall feedrate profile can then be optimized by modulating the knot span durations in an iterative manner where the objective of the feed optimization is basically minimizing the total cycle time without violating any of the axis limits. The five-axis machine tool kinematic transformation is used to obtain the individual axis velocity, acceleration and jerk profiles on
the path. As a result of the iterative optimization process, a time optimal feedrate profile is generated that respects the velocity, acceleration and jerk limitations of the servo drives. The overall scheme of the feedrate profiling is shown in Fig. 4.6.

### 4.3.1. Minimum Jerk Feed Profile

As mentioned in the previous section, the complete $b$-spline can be partitioned into $N_{s}=n-p+1$ parameter segments with arc lengths of $S_{0}, \ldots S_{k} \ldots, S_{N_{s}-1}$. The spline feed profile is then generated through the specified time durations, $T_{0}, \ldots, T_{k}, \ldots, T_{N_{s}-1}$ required to travel the corresponding knot span segment distances. The overall $\mathrm{C}^{3}$ displacement profile fitted on the toolpath as a function time $s=s(t)$, and the resulting kinematic profiles are shown in Fig 4.7. The displacement profile for the $\mathrm{k}^{\text {th }}$ knot span segment of the path is defined by a quintic polynomial of time as:

$$
\begin{equation*}
s_{k}(\tau)=A_{k} \tau^{5}+B_{k} \tau^{4}+C_{k} \tau^{3}+D_{k} \tau^{2}+E_{k} \tau+F_{k} \quad, 0 \leq \tau \leq T_{k} . \tag{4.30}
\end{equation*}
$$

The resulting velocity $(\dot{s})$, tangential acceleration ( $\ddot{s}$ ) and jerk ( $\dddot{s}$ ) profiles are obtained simply by differentiation with respect to $\tau$;

$$
\left.\begin{array}{l}
\dot{s}_{k}(\tau)=5 A_{k} \tau^{4}+4 B_{k} \tau^{3}+3 C_{k} \tau^{2}+2 D_{k} \tau+E_{k}  \tag{4.31}\\
\dddot{s}_{k}(\tau)=20 A_{k} \tau^{3}+12 B_{k} \tau^{2}+6 C_{k} \tau+2 D_{k} \\
\dddot{s}_{k}(\tau)=60 A_{k} \tau^{2}+24 B_{k} \tau+6 C_{k}
\end{array}\right\} .
$$

In order to generate the jerk continuous $\left(\mathrm{C}^{3}\right)$ feedrate profile, the piecewise fifth degree displacement profiles for consecutive knot spans segments are connected to each other by imposing the position (s), velocity ( $\dot{s}$ ), acceleration ( $\dddot{s}$ ) and jerk ( $\dddot{s}$ ) boundary conditions,

$$
\left.\left.\begin{array}{c}
s_{k}\left(T_{k}\right)=s_{k+1}(0)  \tag{4.32}\\
\dot{s}_{k}\left(T_{k}\right)=\dot{s}_{k+1}(0) \\
\ddot{s}_{k}\left(T_{k}\right)=\ddot{s}_{k+1}(0) \\
\dddot{s}_{k}\left(T_{k}\right)=\dddot{s}_{k+1}(0)
\end{array}\right\} \Rightarrow \begin{array}{ll}
A_{k} T_{k}^{5}+B_{k} T_{k}^{4}+C_{k} T_{k}^{3}+D_{k} T_{k}^{2}+E_{k} T_{k}+F_{k} & =F_{k+1} \\
5 A_{k} T_{k+1}^{4}+4 B_{k+1} T_{k+1}^{3}+3 C_{k} T_{k+1}^{2}+2 D_{k} T_{k}+E_{k} & =E_{k+1} \\
20 A_{k} T_{k}^{3}+12 B_{k} T_{k}^{2}+6 C_{k} T_{k}+2 D_{k} & =2 D_{k+1} \\
60 A_{k} T_{k}^{2}+24 B_{k} T_{k}+6 C_{k} & =6 C_{k+1}
\end{array}\right\} .
$$



Figure 4.7: Kinematic Profiles on the B-spline Toolpath

Furthermore, as indicated in Chapter 3 of the thesis, smoothing of the generated profiles can be achieved by minimizing the overall jerk content. The objective function is the integral square of jerk throughout the whole motion expressed as:

$$
\begin{equation*}
J_{V}=\int_{0}^{T_{\Sigma}}\left[\frac{\mathrm{d}^{3} s(t)}{\mathrm{d} t^{3}}\right]^{2} d t=\int_{0}^{T_{\Sigma}}[\dddot{s}(t)]^{2} d t \tag{4.33}
\end{equation*}
$$

where $T_{\Sigma}=\sum_{k=0}^{N s_{s}-1} T_{k}$ is the total motion time. The overall jerk objective given in Eq.(4.33) is written for $N_{s}$ local objectives corresponding to each segment as,

$$
\begin{equation*}
J_{V}=\underbrace{\int_{0}^{T_{0}}\left[\dddot{s}_{0}(\tau)\right]^{2} d \tau}_{V_{V_{0}}}+\underbrace{\int_{1}^{T_{1}}\left[\dddot{s}_{1}(\tau)\right]^{2} d \tau}_{V_{V_{1}}}+\ldots+\underbrace{\left.\int_{N_{N_{-1}-1}}^{T_{M_{N-1}}}(\tau)\right]^{2} d \tau}_{J_{V_{N-1}}} \tag{4.34}
\end{equation*}
$$

where the substitution of Eq.(4.31) in Eq.(4.34) results;

$$
J_{V_{k}}=\int_{0}^{T_{k}}\left[\begin{array}{l}
3600 A_{k}^{2} \tau^{4}+2880 A_{k} B_{k} \tau^{3}+  \tag{4.35}\\
\left(720 A_{k} C_{k}+576 B_{k}^{2}\right) \tau^{2}+288 B_{k} C_{k} \tau+36 C_{k}^{2}
\end{array}\right] d \tau .
$$

The coefficients of the minimum jerk $C^{3}$ quintic displacement profile ( $A_{k}, B_{k}, C_{k}, D_{k}, E_{k}, F_{k}$ ) segment are obtained by solving the linear quadratic optimization problem similar to the Eq. (4.25), resulting from jerk minimization objective given in Eq.(4.35) and constrained by the continuity conditions ( $\mathrm{C}^{0}, \mathrm{C}^{1}, \mathrm{C}^{2}, \mathrm{C}^{3}$ ) expressed by Eq.(4.32). The complete formulation and generation of the minimum jerk displacement profile is presented in Appendix A of the thesis.

The minimum jerk $C^{3}$ displacement profile is obtained after the computation of the coefficients. The objective of the feedrate optimization is then minimizing the total motion time ( $T_{\Sigma}$ ) within the kinematic limits of the machine tool. In order to optimize the total cycle time, a second cost function is formulated as the sum of the feed spline motion durations:

$$
\begin{equation*}
T_{\Sigma}=\sum_{k=0}^{N_{s}-1} T_{k} . \tag{4.36}
\end{equation*}
$$

### 4.3.2. Velocity, Acceleration and Jerk Constraints

Constraints on the motion duration cost function in Eq.(4.36) are defined with respect to the kinematic limits of the five-axis machine tool's drives. For instance, the velocity constraints are defined to indicate the maximum allowable speed of each axis and the given tangential feedrate. In addition, in 5-Axis machine tool structures, the rotary axes have relatively low acceleration thresholds in contrast to Cartesian axes, which can cause actuator saturations at sharp curvatures and orientation changes. Therefore, acceleration limits are imposed to constrain the maximum reachable feed on the given toolpath. The jerk limits help to contain the frequency content of acceleration below the natural modes of the feed drives avoiding unwanted structural vibrations as well as excessive tracking error.

### 4.3.3. Inverse Kinematics of a Rotary Tilting Table 5-Axis Machine Tool

The tool tip position $\left[\begin{array}{lll}x^{t} & y^{t} & z^{t}\end{array}\right]^{\mathrm{T}}$ and tool orientation $\left[\begin{array}{lll}O_{x} & O_{y} & O_{z}\end{array}\right]^{\mathrm{T}}$ vectors are obtained along the double b -spline toolpath with respect to the generated position profile, $s=s(t)$. Following that, the structure dependent inverse kinematics transformation must be employed to generate the Cartesian and rotary axis reference commands from position and orientation commands in real time. Commonly, five-axis machine tools have a kinematic structure based on a 3-Axis Cartesian machine with addition of two rotary axes as shown in Fig.4.8. The two rotary axes accomplish the desired orientation and the Cartesian axes perform the linear motion of the tool to the desired location.

The five-axis machine tool configuration to be modelled is illustrated in Fig.4.8. It has a rotary-tilting table placed on the top of sliding the $x-y$ table. The $x-y$ table shows intended motion only along the X and Y directions, leaving the spindle to move the tool up and down. By utilizing a rotary-tilting table, two rotational degrees of freedom are added to the structure. The notation for this configuration is as follows: the rotary axis indicated with $A$
(tilting axis) introduces a rotational degree of freedom around the X axis. On the top of it, the $C$ (rotary axis) axis is located that introduces another rotational degree of freedom around the Z axis (tool axis). Since the cutting tool is symmetric about the tool frame, a five-axis machine tool can achieve any demanded orientation.


Figure 4.8: Kinematic Chain of the Rotary Tilting Table 5-Axis Machine Tool

In order to define the inverse kinematics transformation of this structure, the decoupling method [41] is used, and rotary joint angles are derived by the aid of homogeneous transformation matrices [51]. Firstly, frames (coordinates systems) are assigned for machine tool's elements such as the table, spindle, base and cutting tool shown (See Fig.4.8). The frame $S$ is assigned for the sliding $x-y$ table, which is connected to the base of the machine tool with a prismatic joint. The frame $T$ is assigned for the tilting axis (A), which is a rotational joint fixed to the $x-y$ table. Similarly, $R$ is assigned for the rotary axis $(C)$ attached to the tilting axis. $W_{p}$ is assigned for the workpiece sitting on the rotary table. The frame $S P$
is assigned to the spindle that is connected to the column of the machine tool with a prismatic joint. It has a degree of freedom only along the Z direction. Another frame, OT is defined for the cutting tool. Finally, a fixed reference coordinate system, Ref is placed on the machine tool's base, which is rigidly connected to the column.


Figure 4.9: a) Translation Motion --- b) Rotational Motion

The rotational and translational relationship between these individual frames is presented using homogeneous transformation matrices. For example; pure translations along X, Y and Z directions at the amount of $\delta x, \delta y, \delta z$ between two frames is given by a homogeneous transformation matrix as,

$$
H_{\text {transsation }}=\left[\begin{array}{cccc}
1 & 0 & 0 & X+\delta x  \tag{4.37}\\
0 & 1 & 0 & Y+\delta y \\
0 & 0 & 1 & Z+\delta z \\
0 & 0 & 0 & 1
\end{array}\right]
$$

where $\mathrm{X}, \mathrm{Y}, \mathrm{Z}$ are the initial offset values between the origins of frames. Considering Eq.(4.37), the pure translation is illustrated in Fig.4.9 (a). The machining point is defined by the vector $P=\left[\begin{array}{lll}x^{t} & y^{t} & z^{t}\end{array}\right]^{\mathrm{T}}$ with respect to the workpiece frame $\left(W_{p}\right)$ that is located on the
$x-y$ table. Using Eq.(4.37), the transformed tool tip vector is represented with respect to the reference (Ref) coordinate system as:

$$
\left[\begin{array}{c}
P^{\prime}  \tag{4.38}\\
1
\end{array}\right]=H_{\text {translation }} \cdot\left[\begin{array}{c}
x^{\prime} \\
y^{t} \\
z^{t} \\
1
\end{array}\right] .
$$

Similarly, if the workpiece is located on the rotating table of the machine tool, the relationship between the reference and the workpiece frames is given by a pure rotation matrix. As shown in Fig 4.9 (b), a rotation around Z axis of the workpiece coordinate system by the angle of $\delta \theta_{z}$ can be represented by a pure $3 \times 3$ rotation matrix as,

$$
R=\left[\begin{array}{ccc}
\cos \left(\theta_{z}+\delta \theta_{z}\right) & -\sin \left(\theta_{z}+\delta \theta_{z}\right) & 0  \tag{4.39}\\
-\sin \left(\theta_{z}+\delta \theta_{z}\right) & \cos \left(\theta_{z}+\delta \theta_{z}\right) & 0 \\
0 & 0 & 1
\end{array}\right]
$$

where $\theta_{z}$ is the initial relative rotation between the frames. This rotation can also be defined by the augmented $4 \times 4$ homogeneous transformation matrix as:

$$
H_{\text {rotation }}=\left[\begin{array}{cccc}
\cos \left(\theta_{z}+\delta \theta_{z}\right) & -\sin \left(\theta_{z}+\delta \theta_{z}\right) & 0 & 0  \tag{4.40}\\
\sin \left(\theta_{z}+\delta \theta_{z}\right) & \cos \left(\theta_{z}+\delta \theta_{z}\right) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array}\right] .
$$

After the rotation is performed, the transformed tool tip vector in the reference coordinate system is expressed as:

$$
\left[\begin{array}{c}
P^{\prime}  \tag{4.41}\\
1
\end{array}\right]=H_{\text {rotation }} \cdot\left[\begin{array}{c}
x^{t} \\
y^{t} \\
z^{t} \\
1
\end{array}\right]
$$

Respectively, intended rotations around X and Y axes are represented by similar homogeneous transformation matrices [51]. By the current frame approach, the homogeneous transformation matrices can be multiplied in a sequence in order to express the relative motion of the end frame with respect to any given reference frame. For instance, if N rigid bodies are connected in series and the corresponding homogenous transformation matrices between connecting axes are known, relation between the last coordinate system ( $\mathrm{N}^{\text {th }}$ frame) in terms of the reference coordinate system is the sequential product of all of the homogenous transformation matrices:

$$
\begin{equation*}
H_{N}^{0}=\prod_{m=1}^{N} H_{m}^{m-1}=H_{1}^{0} H_{2}^{1} H_{3}^{2} H_{4}^{3} \ldots H_{N}^{N-1} . \tag{4.42}
\end{equation*}
$$

### 4.3.3.1 Rotational Motions

The objective in five-axis machining is to keep the tool axis always aligned to the specified orientation and the tool tip placed at the reference machining point. In this modelled configuration, two rotary axes rotate the workpiece to align the orientation vector with the tool axis. Hence, the rotary axis rotation values are first calculated to establish the correct orientation of the tool.


Figure 4.10: a) Table Orientation Before Cutting --- b) Table Orientation During Cutting

In Fig. 4.10, the rotational motion of the table is shown. During the machining operation, the workpiece is continuously rotated to align the orientation vector $O(u)=\left[O_{x}(u) \quad O_{y}(u) \quad O_{z}(u)\right]^{T}$ with respect to workpiece frame to the given tool orientation vector $O^{T o o l}$. The tool orientation is already fixed as $O^{T o o l}=\left[\begin{array}{lll}0 & 0 & 1\end{array}\right]^{T}$ along the Z axis with respect to the reference coordinate system. As shown in Fig. 4.10 (b), two rotations are performed in a sequence to accomplish this task. The first rotation is carried out by the tilting axis $A$ by the angle of $\theta_{A}$ and followed by the rotary axis $C$ by the amount of $\theta_{C}$. The rotation of the $A$ axis is given by a homogeneous transformation matrix as:

$$
H_{\theta_{A}}=\left[\begin{array}{cccc}
1 & 0 & 0 & 0  \tag{4.43}\\
0 & \cos \left(\theta_{A}\right) & -\sin \left(\theta_{A}\right) & 0 \\
0 & \sin \left(\theta_{A}\right) & \cos \left(\theta_{A}\right) & 0 \\
0 & 0 & 0 & 1
\end{array}\right],
$$

and the pure rotation of the $C$ axis can be written as:

$$
H_{\theta_{C}}=\left[\begin{array}{cccc}
\cos \left(\theta_{C}\right) & -\sin \left(\theta_{C}\right) & 0 & 0  \tag{4.44}\\
\sin \left(\theta_{C}\right) & \cos \left(\theta_{C}\right) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array}\right] .
$$

Hence, the relative rotation between the workpiece and the reference frames is composed of these two sequential rotations. In order to achieve the correct orientation, the representation of the orientation vector respect to the reference coordinate system must equal to the fixed orientation of the tool vector. Using the current frame approach, the transformation matrices (Eq. (4.43) and Eq. (4.44) ) are multiplied in a sequence resulting in the following motion equation:

$$
\begin{equation*}
H_{\theta_{A}} \cdot \dot{H}_{\theta_{c}} \cdot O=O^{T o o l} . \tag{4.45}
\end{equation*}
$$

Eq. (4.45) cän be written as a set of equations in the matrix form as:

$$
\left[\begin{array}{cccc}
1 & 0 & 0 & 0  \tag{4.46}\\
0 & \cos \left(\theta_{A}\right) & -\sin \left(\theta_{A}\right) & 0 \\
0 & \sin \left(\theta_{A}\right) & \cos \left(\theta_{A}\right) & 0 \\
0 & 0 & 0 & 1
\end{array}\right] \cdot\left[\begin{array}{cccc}
\cos \left(\theta_{C}\right) & -\sin \left(\theta_{C}\right) & 0 & 0 \\
\sin \left(\theta_{C}\right) & \cos \left(\theta_{C}\right) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array}\right] \cdot\left[\begin{array}{c}
O_{x}(u) \\
O_{y}(u) \\
O_{z}(u) \\
1
\end{array}\right]=\left[\begin{array}{c}
O^{\text {Tool }} \\
1
\end{array}\right]=\left[\begin{array}{c}
0 \\
0 \\
1 \\
1
\end{array}\right] .
$$

Rotation angles $\theta_{A}$ and $\theta_{C}$ are obtained by solving the first two rows of Eq. (4.46), and this decoupling method gives solutions for the rotary axes shown in Table 4.1.

Table 4.1 Rotary Axis Joint Angles

|  | $O_{y}<0$ | $O_{y}=0$ |  |  | $O_{y}>0$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  | $\forall O_{x}$ | $O_{x}<0$ | $O_{x}=0$ | $O_{x}>0$ | $\forall O_{x}$ |
| $\theta_{C}(u)$ | $\tan ^{-1}\left(\frac{O_{x}(u)}{O_{y}(u)}\right)+\pi$ | $-\pi / 2$ | Undefined | $\pi / 2$ | $\tan ^{-1}\left(\frac{O_{x}(u)}{O_{y}(u)}\right)$ |
| $\theta_{A}(u)$ | $\tan ^{-1}\left(\frac{O_{x}^{2}(u)+O_{y}^{2}(u)}{O_{z}(u)}\right)$ | $\tan ^{-1}\left(\frac{O_{x}^{2}(u)+O_{y}^{2}(u)}{O_{z}(u)}\right)$ |  | $\tan ^{-1}\left(\frac{O_{x}^{2}(u)+O_{y}^{2}(u)}{O_{z}(u)}\right)$ |  |

### 4.3.3.2 Cartesian Motions:

Following the rotary joint angles, the Cartesian axis position commands are calculated to position the tool tip at the specified machining point. Homogeneous transformation matrices are defined to express the intended translations and the fixed offsets between the assigned frames. The time varying homogeneous transformation matrix between reference and $x-y$ table frames can be written as,

$$
H_{S}^{\text {Ref }}=\left[\begin{array}{cccc}
1 & 0 & 0 & X(u)+S_{x}  \tag{4.47}\\
0 & 1 & 0 & Y(u)+S_{y} \\
0 & 0 & 1 & S_{z} \\
0 & 0 & 0 & 1
\end{array}\right]
$$

where $X(\mathrm{u})$ and $Y(\mathrm{u})$ are time varying $(u=u(t))$ position of $\mathrm{x}-\mathrm{y}$ table frame origin from the reference frame origin. $S_{x}, S_{y}$, and $S_{z}$ are the initial offset values between those frames, and can be selected depending on the initial position.

Similarly, a fixed homogeneous transformation matrix is defined between table frame and tilting axis frame origins in terms of the offset values $T_{x}, T_{y}, T_{z}$ :

$$
H_{T}^{S}=\left[\begin{array}{cccc}
1 & 0 & 0 & T_{x}  \tag{4.48}\\
0 & 1 & 0 & T_{y} \\
0 & 0 & 1 & T_{z} \\
0 & 0 & 0 & 1
\end{array}\right]
$$

Another homogeneous transformation matrix between tilting and rotary axis frames is written as,

$$
H_{R}^{T}=\left[\begin{array}{cccc}
1 & 0 & 0 & R_{x}  \tag{4.49}\\
0 & 1 & 0 & R_{y} \\
0 & 0 & 1 & R_{z} \\
0 & 0 & 0 & 1
\end{array}\right]
$$

where $R_{x}, R_{y}, R_{z}$ are fixed offsets between frame origins. The workpiece is located on the rotary table. The transformation matrix between rotary axis and workpiece frame origins is similarly given by:

$$
H_{w_{p}}^{R}=\left[\begin{array}{cccc}
1 & 0 & 0 & W p_{x}  \tag{4.50}\\
0 & 1 & 0 & W p_{y} \\
0 & 0 & 1 & W p_{z} \\
0 & 0 & 0 & 1
\end{array}\right]
$$

where $W_{p x}, W_{p y}$ and $W_{p z}$ are the offsets. The transformation matrix between the spindle and tool frame is defined as:

$$
H_{o r}^{S P}=\left[\begin{array}{cccc}
1 & 0 & 0 & O T_{x}  \tag{4.51}\\
0 & 1 & 0 & O T_{y} \\
0 & 0 & 1 & O T_{z} \\
0 & 0 & 0 & 1
\end{array}\right]
$$

where $O T_{x}, O T_{y}$ and $O T_{z}$ are the tool offsets. At last the homogeneous transformation matrix between spindle and the reference frame is given as:

$$
H_{S P}^{\text {Ref }}=\left[\begin{array}{cccc}
1 & 0 & 0 & S P_{x}  \tag{4.52}\\
0 & 1 & 0 & S P_{y} \\
0 & 0 & 1 & Z(u)+S P_{z} \\
0 & 0 & 0 & 1
\end{array}\right]
$$

where $Z(\mathbf{u})$ is the varying spindle frame origin's distance from the reference coordinate system. $S P_{x}, S P_{y}$ and $S P_{z}$ are the initial offset values.

The resultant homogeneous transformation matrix between the fixed reference and varying spindle frames is written by the use of Eq.(4.51) and Eq.(4.52) as:

$$
\begin{equation*}
H_{o T}^{R e f}=H_{S P}^{R e f} \cdot H_{O T}^{S P} . \tag{4.53}
\end{equation*}
$$

Similarly, the homogeneous transformation matrix between the workpiece and reference frames can be written by employing the transformation matrices from Eq. (4.47) to Eq. (4.50), and Eq. (4.43), Eq. (4.44) in a sequence starting from reference to workpiece frames as:

$$
\begin{equation*}
H_{W_{p}}^{R e f}=H_{S}^{\text {Ref }} \cdot H_{T}^{s} \cdot H_{\theta_{A}} \cdot H_{R}^{T} \cdot H_{\theta_{c}} \cdot H_{W_{p}}^{R} . \tag{4.54}
\end{equation*}
$$

The correct positioning of the tool tip to the given machining point is achieved by the motion of $x-y$ table and the spindle. Consequently, the tool tip position is given in tool frame as $P^{T o o l}=\left[\begin{array}{lll}0 & 0 & 0\end{array}\right]^{T}$, and it is written with respect to the reference coordinate frame using Eq.(4.53). The machining point $P=\left[\begin{array}{lll}x^{t}(u) & y^{t}(u) & z^{t}(u)\end{array}\right]^{T}$ is also transformed from workpiece coordinate system to the reference coordinate system using Eq.(4.54).


Figure 4.11: Kinematic Transformation

Hence, the correct positioning is achieved as illustrated in Fig 4.11 by matching the representation of these two vectors with respect to the reference frame using Eq.(4.53) and Eq.(4.54) as:

$$
\underbrace{H_{W_{p}}^{\text {Ref }} \cdot\left[\begin{array}{c}
x^{t}(u)  \tag{4.55}\\
y^{t}(u) \\
z^{t}(u) \\
1
\end{array}\right]}_{\text {Machinig Position Vector }}=\underbrace{H_{o r}^{\text {Ref }} \cdot\left[\begin{array}{l}
0 \\
0 \\
0 \\
1
\end{array}\right] .}_{\text {Tool Tip Position Vector }}
$$

The solution of Eq (4.55) yields the resultant equations for the $\mathrm{X}, \mathrm{Y}, \mathrm{Z}$ Cartesian axis commands:

$$
\begin{align*}
X(u)= & O T_{x}+S P_{x}-R_{x}-T_{x}-S_{x}-\cos \left(\theta_{c}(u)\right) \cdot x^{t}(u)+\sin \left(\theta_{c}(u)\right) \cdot y^{t}(u)-\cos \left(\theta_{c}(u)\right) \cdot W p_{x} \\
& +\sin \left(\theta_{c}(u)\right) W p_{y} \\
Y(u)= & O T_{y}+S P_{y}-T_{y}-S_{y}-\cos \left(\theta_{a}(u)\right) \cdot \sin \left(\theta_{c}(u)\right) \cdot x^{t}(u)-\cos \left(\theta_{a}(u)\right) \cdot \cos \left(\theta_{c}(u)\right) \cdot y^{t}(u)+ \\
& \sin \left(\theta_{a}(u)\right) \cdot x^{t}(u)-\cos \left(\theta_{a}(u)\right) \cdot \sin \left(\theta_{c}(u)\right) \cdot W p_{x}-\cos \left(\theta_{a}(u)\right) \cdot \cos \left(\theta_{c}(u)\right) \cdot W p_{y} \\
& +\sin \left(\theta_{a}(u)\right) \cdot W p_{z}-\cos \left(\theta_{a}(u)\right) \cdot R_{y}+\sin \left(\theta_{a}(u)\right) \cdot R_{z} \\
Z(u)= & -O T_{z}-S P_{z}+T_{z}+S_{z}+\sin \left(\theta_{a}(u)\right) \cdot \sin \left(\theta_{c}(u)\right) \cdot x^{t}(u)+\sin \left(\theta_{a}(u)\right) \cdot \cos \left(\theta_{c}(u)\right) \cdot y^{t}(u) \\
+ & \cos \left(\theta_{a}(u)\right) \cdot z^{t}(u)+\sin \left(\theta_{a}(u)\right) \cdot \sin \left(\theta_{c}(u)\right) \cdot W p_{x}+\sin \left(\theta_{a}(u)\right) \cdot \sin \left(\theta_{c}(u)\right) \cdot W p_{y} \\
+ & \cos \left(\theta_{a}(u)\right) \cdot W p_{z}+\sin \left(\theta_{a}(u)\right) \cdot R_{y}+\cos \left(\theta_{a}(u)\right) \cdot R_{z} \tag{4.56}
\end{align*}
$$

### 4.3.4. Formulation of the Constraints

The Joint variable vector, $M(u)=\left[\begin{array}{llll}X(u) & Y(u) & Z(u) & \theta_{A}(u)\end{array} \theta_{C}(u)\right]^{T}$ is obtained after the inverse kinematics transformation using Eq.(4.46) and Eq.(4.56). Following that, the axis velocity, acceleration, and jerk profiles $(\dot{M}(t), \ddot{M}(t), \ddot{M}(t))$ with respect to the displacement profile $s=s(t)$ are expressed as,

$$
\left.\begin{array}{l}
\dot{M}(t)=M_{s} \cdot \dot{s}(t) \\
\ddot{M}(t)=M_{s s} \cdot \dot{s}(t)^{2}+M_{s} \cdot \ddot{s}(t)  \tag{4.57}\\
\ddot{M}(t)=M_{s s s} \cdot \dot{s}(t)^{3}+3 M_{s s} \cdot \dot{s}(t) \cdot \ddot{s}(t)+M_{s} \cdot \dddot{s}(t)
\end{array}\right\}
$$

where $M_{s,} M_{s s}, M_{s s s}$ are the geometric derivatives of the toolpath computed over the axis reference commands by

$$
\left.\begin{array}{l}
M_{s}(s)=\frac{\mathrm{d} M}{\mathrm{~d} s}=\frac{\mathrm{d} M}{\mathrm{~d} u} \frac{\mathrm{~d} u}{\mathrm{~d} s} \\
M_{s s}(s)=\frac{\mathrm{d}^{2} M}{\mathrm{~d} s^{2}}=\frac{\mathrm{d}^{2} M}{\mathrm{~d} u^{2}}\left(\frac{\mathrm{~d} u}{\mathrm{~d} s}\right)^{2}+M_{s} \frac{\mathrm{~d}^{2} u}{\mathrm{~d} s^{2}}  \tag{4.58}\\
M_{s s s}(s)=\frac{\mathrm{d}^{3} M}{\mathrm{~d} s^{3}}=\frac{\mathrm{d}^{3} M}{\mathrm{~d} u^{3}}\left(\frac{\mathrm{~d} u}{\mathrm{~d} s}\right)^{3}+3 M_{s s} \frac{\mathrm{~d} u \mathrm{~d}^{2} u}{\mathrm{~d} s} \frac{M_{s s s}}{\mathrm{~d} s^{2}} \frac{\mathrm{~d}^{3} u}{\mathrm{~d} s^{3}}
\end{array}\right\} .
$$

For the optimization process, the geometric derivatives of the toolpath are calculated once at the initialization stage by interpolating the toolpath with constant arc increments $(\Delta s)$, stored in a look up table and retrieved during the optimization process for faster evaluation (See Fig.4.6). The limits for tangential feedrate $f_{\max }$, axis velocity $\left(v_{X \text { max }}, v_{Y_{\text {max }}}, v_{Z \text { max }}, v_{\theta_{A} \max }, v_{\theta_{C} \max }\right)$, acceleration $\left(a_{X \text { max }}, a_{Y_{\text {max }}}, a_{Z \text { max }}, a_{\theta_{A} \max }, a_{\theta_{C} \max }\right)$, and jerk $\left(j_{X_{\text {max }}}, j_{Y_{\text {max }}}, j_{Z_{\text {max }}}, j_{\theta_{A} \max }, j_{\theta_{C} \max }\right)$, are the nonlinear constraints to be imposed during the optimization process expressed as:

$$
\begin{align*}
& 0 \leq \dot{s}(t) \leq f_{\max } \\
& -v_{X \text { max }} \leq \dot{X}(t) \leq v_{X \text { max }}, \quad-a_{X \text { max }} \leq \ddot{X}(t) \leq a_{X \text { max }}, \quad-j_{X \text { max }} \leq \dddot{X}(t) \leq j_{X \text { max }} \\
& -v_{Y \text { max }} \leq \dot{Y}(t) \leq v_{Y \text { max }}, \quad-a_{Y \text { max }} \leq \ddot{Y}(t) \leq a_{Y \text { max }}, \quad-j_{Y \text { max }} \leq \dddot{Y}(t) \leq j_{Y \text { max }}  \tag{4.59}\\
& -v_{Z_{\text {max }}} \leq \dot{Z}(t) \leq v_{Z \text { max }}, \quad-a_{Z_{\text {max }}} \leq \ddot{Z}(t) \leq a_{Z \text { max }}, \quad-j_{Z_{\text {max }}} \leq \dddot{Z}(t) \leq j_{Z_{\text {max }}} \\
& -v_{\theta_{A} \max } \leq \dot{\theta}_{A}(t) \leq v_{\theta_{A} \max }, \quad-a_{\theta_{A} \max } \leq \ddot{\theta}_{A}(t) \leq a_{\theta_{A} \max }, \quad-j_{\theta_{1} \max } \leq \dddot{\theta}_{A}(t) \leq j_{\theta_{A} \max } \\
& \left.-v_{\theta_{c} \max } \leq \dot{\theta}_{C}(t) \leq v_{\theta_{c} \max }, \quad-a_{\theta_{c} \max } \leq \ddot{\theta}_{C}(t) \leq a_{\theta_{c} \max }, \quad-j_{\theta_{C} \max } \leq \dddot{\theta}_{C}(t) \leq j_{\theta_{c} \max }\right)
\end{align*}
$$

for $0 \leq t \leq T_{\Sigma}$. The above constraints are normalized and separated into individual inequalities,

$$
\begin{align*}
& C_{f}^{+}(t)=\dot{s}(t) / f_{\text {max }}-1 \leq 0, \quad C_{f}^{-}(t)=-\dot{s}(t) / f_{\text {max }} \leq 0 \\
& C_{\dot{X}}^{+}(t)=\dot{X}(t) / v_{X \text { max }}-1 \leq 0, C_{\dot{X}}^{-}(t)=-\dot{X}(t) / v_{X \text { max }}-1 \leq 0 \\
& C_{\dot{Y}}^{+}(t)=\dot{Y}(t) / v_{Y_{\text {max }}}-1 \leq 0, \quad C_{\dot{Y}}^{-}(t)=-\dot{Y}(t) / v_{Y_{\text {max }}}-1 \leq 0 \\
& C_{\dot{Z}}^{+}(t)=\dot{Z}(t) / v_{Z \text { max }}-1 \leq 0, \quad C_{\dot{Z}}^{-}(t)=-\dot{Z}(t) / v_{Z \max }-1 \leq 0 \\
& C_{\dot{\theta}_{A}}^{+}(t)=\dot{\theta}_{A}(t) / v_{\theta_{A} \max }-1 \leq 0, \quad C_{\dot{\theta}_{A}}^{-}(t)=-\dot{\theta}_{A}(t) / v_{\theta_{A} \max }-1 \leq 0 \\
& \left.C_{\dot{\theta}_{C}}^{+}(t)=\dot{\theta}_{C}(t) / v_{\theta_{c} \max }-1 \leq 0, \quad C_{\dot{\theta}_{C}}^{-}(t)=-\dot{\theta}_{C}(t) / v_{\theta_{c} \max }-1 \leq 0\right\} \\
& C_{\ddot{X}}^{+}(t)=\ddot{X}(t) / a_{X \text { max }}-1 \leq 0, \quad C_{\ddot{X}}^{-}(t)=-\ddot{X}(t) / a_{X \text { max }}-1 \leq 0 \\
& C_{\dot{Y}}^{+}(t)=\ddot{Y}(t) / a_{Y \text { max }}-1 \leq 0, \quad C_{\ddot{Y}}^{-}(t)=-\ddot{Y}(t) / a_{Y_{\text {max }}}-1 \leq 0 \\
& C_{\dot{Z}}^{+}(t)=\ddot{Z}(t) / a_{Z \max }-1 \leq 0, \quad C_{\ddot{Z}}^{-}(t)=-\ddot{Z}(t) / a_{Z \max }-1 \leq 0 \\
& C_{\ddot{\theta}_{A}}^{+}(t)=\ddot{\theta}_{A}(t) / v_{\theta_{A} \max }-1 \leq 0, \quad C_{\ddot{\theta}_{A}}^{-}(t)=-\ddot{\theta}_{A}(t) / v_{\theta_{1} \max }-1 \leq 0 \\
& C_{\ddot{\theta}_{C}}^{+}(t)=\ddot{\theta}_{C}(t) / v_{\theta_{C} \max }-1 \leq 0, \quad C_{\ddot{\theta}_{c}}^{-}(t)=-\ddot{\theta}_{C}(t) / v_{\theta_{c} \max }-1 \leq 0 \\
& C_{\widetilde{X}}^{+}(t)=\dddot{X}(t) / j_{X \text { max }}-1 \leq 0, \quad C_{\widetilde{X}}^{-}(t)=-\dddot{X}(t) / j_{X_{\text {max }}}-1 \leq 0 \\
& C_{\breve{Y}}^{+}(t)=\dddot{Y}(t) / j_{Y \text { max }}-1 \leq 0, \quad C_{\breve{Y}}^{-}(t)=-\dddot{Y}(t) / j_{Y \text { max }}-1 \leq 0 \\
& C_{\tilde{Z}}^{+}(t)=\dddot{Z}(t) / a_{Z \text { max }}-1 \leq 0, \quad C_{\ddot{Z}}^{-}(t)=-\dddot{Z}(t) / a_{Z \text { max }}-1 \leq 0 \quad \text { Jerk Constraints } \\
& C_{\ddot{\theta}_{A}}^{+}(t)=\dddot{\theta}_{A}(t) / v_{\ddot{\theta}_{A} \max }-1 \leq 0, \quad C_{\ddot{\theta}_{A}}^{-}(t)=-\dddot{\theta}_{A}(t) / v_{\ddot{\theta}_{A} \max }-1 \leq 0  \tag{4.60}\\
& \left.C_{\ddot{\theta}_{c}}^{+}(t)=\dddot{\theta}_{C}(t) / v_{\ddot{\theta}_{c} \max }-1 \leq 0, \quad C_{\ddot{\theta}_{C}}^{-}(t)=-\dddot{\theta}_{C}(t) / v_{\ddot{\theta}_{c} \max }-1 \leq 0\right]
\end{align*}
$$

The inequality constrains in Eq. (4.60) are grouped together and written in the vector form of

$$
C(t)=\left[\begin{array}{llllllll}
C_{f}^{+}(t) & C_{\dot{x}}^{+}(t) \ldots & C_{\dot{X}}^{+}(t) \ldots & C_{\tilde{X}}^{+}(t) \ldots & \vdots & C_{f}^{-}(t) & C_{\tilde{X}}^{-}(t) \ldots & C_{\dot{x}}^{-}(t) \ldots
\end{array} C_{\tilde{X}}^{-}(t) \ldots\right]^{T},(5.61)
$$

and formulated for the optimization process as:

$$
\begin{equation*}
\left.C(t) \leq 0 \quad \text { for } \quad 0 \leq t \leq T_{\Sigma}\right\} . \tag{4.62}
\end{equation*}
$$

### 4.3.5. Solution of the Optimization Problem

The optimization problem is given by Eq. (4.36) and Eq. (4.62):

$$
\begin{equation*}
\min _{T_{k}} T_{\Sigma}=\min _{T_{k}} \sum_{k=0}^{N s-1} T_{k} \text { subject to } C(t) \leq 0 \text { for } 0 \leq t \leq T_{\Sigma} \tag{4.63}
\end{equation*}
$$

which is a linear programming problem with non-linear constraints because of the fact that the derivative of the cost function respect to the optimization variables are linear. It is solved by MATLAB's Optimization toolbox. Each segment duration is divided into ( $\sim 5 \ldots 100$ ) subdivisions proportional to its corresponding arc length $S_{k}$ as shown in Fig. 4.12 and the look-up table for the geometric derivatives is prepared by Eq. (4.58).


Figure 4.12: Constraint Evaluation

During the optimization process, the axis kinematic profiles are calculated using Eq. (4.57) over the look-up table and the constraints are checked at fixed subdivisions using Eq. (4.0) of the segment durations. As the segment durations are modulated, the number of constraints to be checked remains the same, but the evaluation locations change since the subdivisions are stretched out or compressed. Such a 'floating constraint' scheme allows analytical computation of the kinematic profile gradients with respect to the optimization variables $(\partial s / \partial \lambda, \partial \dot{s} / \partial \lambda, \partial \ddot{s} / \partial \lambda, \partial \ddot{s} / \partial \lambda)$, as well as the constraint gradient $(\partial \mathrm{C}(t) / \partial \lambda)$, which enhances the convergence of the optimization given in Appendix B.

### 4.4. Simulation Results

### 4.4.1. Comparison of the Interpolation Methods

Four real time interpolation techniques of the $b$-spline toolpath are compared. The feedrate consistency and fluctuations are observed for the following interpolation methods:
(1) $1^{\text {st }}$ order Taylor's Approximation
(2) $2^{\text {nd }}$ order Taylor's Approximation
(3) Feed correction polynomial interpolation
(4) Feed correction with iterative interpolation

In the literature, the interpolation of $b$-spline toolpaths is performed using Taylor series approximations [74, 87, 92, 93]. Therefore, the performance of the feed correction polynomial interpolation technique is compared to $1^{\text {st }}$ and $2^{\text {nd }}$ order Taylor's Approximation methods and summarized in Table 4.2. The b-spline toolpath for the tests is shown in Fig. 4.13. It is a round arbitrary designed shape with sharp changes in the curvature near quadrants.

The toolpath is run at constant feedrate of $100[\mathrm{~mm} / \mathrm{sec}]$ and interpolated at 1 [ msec ] sampling. Results are summarized for the velocity of the tool tip in Fig. 4.14. Interpolation result of the toolpath with $1^{\text {st }}$ order Taylor approximation is shown in Fig. 4.14 (a). The fluctuation on the tangential feedrate of the tool tip is observed as maximum 0.47 [\%]. Since calculation of $b$-spline derivatives demand high computational load, each iteration needs $\sim 4$ [msec] on a P4 2.6 GHz desktop computer running MATLAB. However, it can be implemented by a much more efficient computational platform for real time interpolation. In Fig. 4.15 (b) results for the $2^{\text {nd }}$ order Taylor Approximation can be seen. The maximum fluctuation is reduced by 0.017 [\%]. Since more terms are included in the approximation, the resultant time expense is increased to $\sim 6$ [ msec ]. In Fig. 4.14 (c), the same toolpath is interpolated using the feed correction polynomial. The correction polynomial is initially fitted offline and the real - time interpolation is performed without any recursive calculation of the spline parameter. Therefore, the computational load can be neglected ( $<0.02$ [ msec$]$ ). The maximum fluctuation in the feedrate is 0.03 [\%], which is closer to the $2^{\text {nd }}$ Order Taylor
approximation. This indicates that the method can be used for real time b -spline interpolation as compared to computationally expensive Taylor approximation techniques. In the last simulation, the toolpath is interpolated with the recursive interpolation combined with feed correction polynomial guess. There is practically no feedrate fluctuation for the b -spline toolpath. However, at least 2 iterations are needed, which increases the computational time up to $\sim 8$ [ msec ]. In contrast to any Taylor approximation method, it guarantees higher accuracy at for parameter calculation.

Table 4.2 Summary of Interpolation Results

| Interpolation Method | Feed Fluctuation <br> $[\%]$ | Time Expense <br> $[\mathrm{msec}]$ |
| :---: | :---: | :---: |
| $\mathbf{1}^{\text {st }}$ order Taylor Series Apprx. | 0.47 | $\cong \mathbf{4}$ |
| $\mathbf{2}^{\text {nd }}$ order Taylor Series Apprx. | 0.017 | $\cong \mathbf{6}$ |
| Feed Correction Poly. | 0.03 | $<\mathbf{0 . 0 2}$ |
| Feed Correction Poly. with iterative solution | 0.0001 | $\cong \mathbf{8}$ |



Figure 4.13: B-spline Toolpath


Figure 4.14: Feedrate Fluctuations

### 4.4.2. Comparison of Constant and Optimized Feedrate Profiles

The feedrate optimization algorithm is tested on the toolpath that is generated in chapter 3 and shown in Fig. 4.15. It consists of $b$-splines defined by 42 control points which contains mildly varying curvature and orientation at the start section as well as sharp changes towards the end. The simulations are performed with respect to the kinematic limits of the HELLER MC 25 five-axis machining center. The velocity, acceleration and jerk limits are given in the following Table 4.3.

The maximum feedrate limit on the toolpath is set as $f_{\text {max }}=200[\mathrm{~mm} / \mathrm{sec}]$. The optimization of the toolpath required 120 [sec] using a windowing technique [19] at P2.6 Ghz desktop computer running MATLAB and resulted in a motion duration of 2.09 [sec]. Trying to travel the same toolpath without violating any of the axis kinematic limits results in using a feedrate profile of $53[\mathrm{~mm} / \mathrm{sec}]$ with cubic acceleration/decelerations of 1250 $\left[\mathrm{mm} / \mathrm{sec}^{2}\right]$, and a tangential jerk of $15000\left[\mathrm{~mm} / \mathrm{sec}^{3}\right]$ having an overall motion duration of 3.751 [sec]. With the optimized feed profile, the improvement on the total cycle time is 44 [\%]. The feedrate profile is shown in Fig. 4.16, which indicates that the tool tip velocity can be increased up to $200[\mathrm{~mm} / \mathrm{sec}]$ at the start section of the toolpath while taking the advantage of the straight feasible region. On the other hand, it is scheduled conservatively towards the end where the change of tool orientation and curvature is quite sharp. The linear and rotary axis kinematic profiles are shown in Fig. 4.17 and Fig. 4.18 indicating that the major applied limits are the axis acceleration and jerk.

Table 4.3: Kinematic Limits of 5-Axis Machine Tool

| Heller MC25 Axis <br> Limits | Velocity Limits | Acceleration <br> Limits | Jerk Limits |
| :---: | :--- | :---: | :---: |
| Linear <br> Axis(X,Y,Z) | $v_{\operatorname{maz}}=1000 \frac{\mathrm{~mm}}{\mathrm{sec}}$ | $a_{\operatorname{maz}}=10000 \frac{\mathrm{~mm}}{\mathrm{sec}^{2}}$ | $j_{\operatorname{maz}}=30000 \frac{\mathrm{~mm}}{\mathrm{sec}^{3}}$ |
| Rotary Axis (A) | $v_{\theta_{1} \max }=4.2 \frac{\mathrm{rad}}{\mathrm{sec}}$ | $a_{\theta_{A} \max }=6.3 \frac{\mathrm{rad}}{\mathrm{sec}^{2}}$ | $j_{\theta_{A} \max }=62.8 \frac{\mathrm{rad}}{\mathrm{sec}^{3}}$ |
| Rotary Axis (C) | $v_{\theta_{c} \max }=10.5 \frac{\mathrm{rad}}{\mathrm{sec}}$ | $a_{\theta_{C} \max }=17.3 \frac{\mathrm{rad}}{\mathrm{sec}^{2}}$ | $j_{\theta_{C} \max }=251 \frac{\mathrm{rad}}{\mathrm{sec}^{3}}$ |



Figure 4.15: B-spline Toolpath


Figure 4.16: Optimized and Unoptimized Feedrate Profiles


Figure 4.17: Optimized and Unoptimized Cartesian Axis Profiles


Figure 4.18: Optimized and Unoptimized Rotary Axis Profiles

### 4.4.3. Summary

The interpolation of $b$-spline toolpaths and optimal feedrate profiling are investigated in this section. By utilizing the feedrate correction polynomial, it is shown that $b$-spline toolpaths are interpolated in real time efficiently. Feed fluctuations arising from inaccurate calculation of the spline parameter is further eliminated using the recursive interpolation scheme. For 5 - axis machining, the feed correction polynomial interpolation method allows tangential velocity at the tool tip to be kept consistent. However, the effective feedrate is changing along the tool axis because of the angular velocity of tool. Therefore, during flank milling operation, feedrate variations can be encountered along tool axis. In contrast, in 3 -
axis machining this method can easily be employed for interpolation of any $b$-spline and Nurbs toolpaths in order to eliminate unwanted feed fluctuations.

Furthermore, a feedrate scheduling algorithm for 5-Axis machine tool is implemented. The feedrate curve is defined as a fifth order $\mathrm{C}^{3}$ continuous spline providing jerk continuous kinematic profiles to the drives. Furthermore, the optimized feedrate profile intends minimum cycle time while not violating the axes' velocity, acceleration and jerk limits. The algorithm can be used for high speed machining where continuously alteration of the feedrate profile is not crucial for the manufacturing process. For instance, its implementation is suitable for rough machining and pre-finishing of dies and molds combined with the previously presented b -spline toolpath.

## Chapter 5

## Conclusions

### 5.1. Conclusions

In this thesis, a smooth and optimal trajectory generation algorithm is presented for fiveaxis high speed machining. A $b$-spline toolpath has been generated in order to deliver continuous tool motion for five-axis machining by compressing linear G1 blocks assigned in the initial NC program. The real time interpolation of the $b$-spline toolpath is realized usigng the Feed Correction Polynomial method for minimal feedrate fluctuation. A jerk continuous feedrate optimization technique has been formulated for traveling along the $b$-spline toolpath in minimum time while adhering to the machine tool's kinematic limits such as velocity, acceleration and jerk. The thesis has the following contributions:

The spline toolpath generation algorithm employs discrete toolpath position and orientation vectors extracted from the CAM system and generates a smooth toolpath composed of two $b$-splines with respect to the user given position and angular tolerance specifications. The double spline toolpath format is capable of interpolating tool position as well as the orientation vectors continuously in a synchronized way, decreasing the need for a machine tool dependent inverse kinematics post processing. The toolpath format is convenient for five-Axis high speed machining purposes and accepted by conventional CNC systems for real time interpolation.

The smoothness of the generated toolpath is further improved using two optimization procedures. The first linear optimization technique is applied during the least squares fitting of the tool tip position spline, and it modifies the spline for smoother motion of the machine tool's Cartesian axes. However, the linear optimization is not practical when the tolerance specifications are tight. On the other hand, because of the machine tool's kinematic structure, any irregularity in the angular motion of the tool is directly reflected to the complete multiaxis motion. As an alternative the non-linear optimization algorithm addresses the angular motion of the tool, and eliminates any oscillation in the angular acceleration by modifying the orientation spline of the toolpath. The non-linear method the path to be more accurately
tracked given the limited controller bandwidth, since any excitation of the servo and structural dynamics are avoided.

The real time interpolation of the $b$-spline toolpath is improved using Feed Correction Polynomial Method. As compared to the conventional $b$-spline interpolation algorithms based on first order Taylor Series' expansion series; the implemented method is computationally less expensive and provides better feedrate consistency. Furthermore, combined with the iterative parameter correction scheme, the feed correction polynomial method exhibits an order of magnitude less feed fluctuation at steep curvature regions in contrast to second order Taylor's approximation.

The feedrate optimization algorithm utilizes a quintic $\mathrm{C}^{3}$ spline as displacement profile in order to realize a jerk continuous trajectory in five-axis machining. The spline displacement profile is especially constructed using minimum jerk criteria ensuring a smooth and continuous change in the feedrate enabling better tracking performance of the drives. The minimum cycle time objective is injected into the generation of optimal feed profile, and is constrained with the distinctive velocity, acceleration and jerk limits of the cartesian and rotational axes. The minimum cycle time method combines the ability to continuously modulate the feed, acceleration and jerk profiles along the spline toolpath for five-Axis high speed machining, and smoothness nature enhances the tracking performance of the servo control system while taking the advantage of machine tool's physical limits.

### 5.2. Future Research Directions

The trajectory generation techniques presented in this thesis are practically applicable to the state of art five-Axis high speed machining. However, the proposed algorithms are tested only in virtual environment using the kinematic structures of widely used five-Axis machine tools. Experimental validations should to be conducted on physical test platforms, and the resulting contouring performance should be investigated including the tracking errors of each servo system using the nonlinear kinematic transformation between the joint space and workpiece coordinates.

To expand the toolpath fitting technique to surface finish machining operations, limits for the position and orientation errors of the generated toolpath must be readjusted by including
the tool geometry, such as the tool radius of hemispherical or toroidal cutters in order to eliminate any possible interference between the tool and the workpiece. For instance, in the case of toroidal tool, the tool tip position error is coupled to the orientation error.

The feedrate varies along the tool axis because of the angular velocity leading to fluctuations and oscillations in the feed vector during flank milling operations especially where the angular velocity is more dominant then the tool tip velocity. To improve the overall feed consistency, the upper b-spline of the toolpath must be re-parameterized or the interpolation step size is required to be adjusted that will reduce the angular acceleration of the tool.

The feedrate profiling algorithm uses a smooth quintic polynomial spline. However, the replacement of quintic polynomial with a b-spline feed profile would enhance the convergence of the algorithm by utilizing its local modification property as an effective windowing technique. The actuation dynamics should also be included in feedrate optimization, which will prevent exceeding the actuation torques.

## Bibliography

[1] Affouard, A., Duc, E., Lartigue, C., Langeron, J.-M., Bourdet, P., 2004, "Avoiding 5Axis Singularities Using Tool Path Deformation", International Journal of Machine Tools and Manufacture, Vol. 44, No.4, pp. 415-425.
[2] Alhanaty, M., Bercovier, M., 2001, "Curve and Surface Fitting and Design by Optimal Control Methods", Computer Aided Design, Vol. 33, No. 2, pp. 167-182.
[3] Altintas, Y., Erkorkmaz, K., 2003. "Feedrate Optimization for Spline Interpolation in High Speed Machine Tools", Annals of CIRP, Vol. 49, No. 1, pp. 265-270.
[4] Altintas, Y., 2000, Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations and CNC Design, Cambridge University Press.
[5] Bedi, S., Quan, N., 1992, "Spline Interpolation Technique for NC Machines", Computers in Industry, Vol. 18, No. 3, pp. 307-313.
[6] Bobrow, J. E., Dubowsky, S., Gibson, J. S., 1985, "Time-optimal Control of Robotic Manipulators Along Specified Paths", International Journal of Robotics Research, Vol 4, No. 3, pp. 3-17.
[7] Bohez, E.L.J., 2002, "Five-Axis Milling Machine Tool Kinematic Chain Design and Analysis", International Journal of Machine Tools and Manufacture, Vol. 42, No.4, pp. 505-520.
[8] Butler, J., Haack, B., Tomizuka, M., 1998, "Reference Generation for High Speed Corrdinated Motion of a Two Axis System", Symposium of Robotics, ASME Winter Annual Meeting, Chicago, IL, USA, DSC Vol. 11, pp. 457-470.
[9] Cao, B., Dodds, G.I., Irwin, G.W., 1997, " Constrained Time-Efficient and Smooth Cubic Spline Trajectory Generation for Industrial Robots", Proceedings of IEE, Control Theory and Applications, Vol. 144, No. 5, pp. 467-475
[10] Cebula, A. J., Zsombor-Murray, Paul J., 2003, "Nurbs to Avoid Boundary Orientation Poses in Serial Manipulators", Journal of Robotic Systems, Vol. 20, No. 12, pp.723736
[11] Chang, Y.-H., Lee, T.-T., Liu, C.-H., 1988, " On-Line Cartesian Path Trajectory Planning For Robot Manipulators", Proceedings of 1988 IEEE International Conference on Robotics and Automation, pp. 62-67.
[12] Cheng, M.-Y., Tsai, M.-C., Kuo, J.-C., 2002, "Real-time Nurbs Command Generators for CNC Servo Controllers", International Journal of Machine Tools and Manufacture, Vol. 42, No. 7, pp. 801-813.
[13] Chou, J.-J., Yang, D.C.H., 1991, "On the Motion Coordination and Planning of Multi Axis Manipulators with Orthogonal Regional Structures", Fifth International Conference on Advanced Robotics, pp. 1814-1819.
[14] Chou, J.-J., Yang, D.C.H., 1992, "Tool path Geometry for Multi-axis Kinematics Conversion in CAD-CAM Integration", ASME Design Engineering Division, 22nd Biennial Mechanisms Conference, Vol. 46, pp. 519-529.
[15] Chou, J.-J., Yang, D.C.H., 1993, "Automatic Generation of Piecewise Constant Speed Motion for Multi-axis Machines", ASME Design Engineering Division, 22nd Biennial Mechanisms Conference, Vol. 46, pp. 531-536.
[16] Constantinescu, D., Croft, E.A., 2000, " Smooth and Time-optimal Trajectory Planning for Industrial Manipulators Along Specified Paths", Journal of Robotic Systems, Vol. 17, No. 5, pp. 233-249
[17] Elkeran, A., El-Baz, M.A., "Nurbs Feedrate Adaptation for 3-Axis CNC Machining", Maintence Resources, http://www.maintenanceresources.com / referencelibrary/ ezine /nurbs.htm
[18] Erkorkmaz, K., 1999, M.A.Sc. Thesis:" High Speed Contouring Control for Machine Tool Drives", University of British Columbia, Department of Mechanical Engineering, BC, Canada
[19] Erkorkmaz, K., 2004, Ph.D. Thesis:" Optimal Trajectory Generation and Precision Tracking Control for Multi-Axis Machines", University of British Columbia, Department of Mechanical Engineering, BC, Canada
[20] 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, Vol. 41, No. 9, pp.1323-1345
[21] Erkorkmaz, K., Altintas, Y., 2001, "Trajectory Generation for High Speed Milling of Molds and Dies", Proceedings of the 2nd International Conference and Exhibition on Design and Production of Does and Molds, Kusadasi, Turkey, DM_46.
[22] Erkorkmaz, K. Altintas, Y., 2002, "Optimal Spline Interpolation for High Speed Machine Tools", Proceedings of 6th Biennial Conference on Engineering Systems Design and Analysis, Istanbul, Turkey.
[23] Erkorkmaz, K., Altintas, Y., 2003, "Quintic Spline interpolation with Minimal Feed Fluctuation", Proceedings of the ASME Manufacturing Engineering Division, MED-IMECE2003-42428.
[24] Dyllong, E., Visioli, A., 2003, "Planning and Real time Modification of a Trajectory using Spline Techniques", Robotica, Vol. 21, pp.475-482.
[25] Gossard, F.-L., David, C., 1995, "Multidimensional Curve Fitting to Unorganized Data Points by Nonlinear Minimization", Computer Aided Design, Vol. 27, No.1, pp. 48-58.
[26] Farin, G., 1993, Curves and Surfaces for Computer Aided Geometric Design - A Practical Guide, Academic Press.
[27] Farouki, R. T., Shah, S., 1996, "Real-time CNC Interpolators for Pythagoreanhodograph Curves", Computer Aided Geometric Design, Vol. 13, No. 7, pp. 583-600.
[28] Flash, T., Hogan, N., 1985, "The Coordination of Arm Movements: an Experimentally Confirmed Mathematical Model", Journal of Neuroscience, Vol. 5, pp. 1688-1703.
[29] Fleisig, R.V., Spence, A.D., 2001, "Constant Feed and Reduced Angular Acceleration Interpolation Algorithm for Multi-axis Machining", Computer Aided Design, Vol. 33, No.1, pp. 1-15.
[30] Goldenthal, R., Bercovier, M., 2004, "Spline Curve Approximation and Design by Optimal Control Over the Knots", Computing, Vol. 72, No. 1-2, pp.53-64.
[31] Hohenberger, W., Reuding, T., 1995, "Smoothing Rational B-spline Curves Using the Weights in an Optimization Procedure", Computer Aided Geometric Design, Vol.12, No. 8, pp. 837-848
[32] Horn, B. K. P., 1983, "Curve of Least Energy", ACM Transactions on Mathematical Software, Vol. 9, No.3, pp.441-460.
[33] Jee, S., Koo, T., 2003, "Toolpath Generation for NURBS Surface Machining", Proceedings of the American Control Conference, Vol. 3, pp. 2614-2619.
[34] Jensen, C.G., Anderson, D.C., 1992, "Accurate Tool Placement and Orientation for Finish Surface Machining", ASME Production Engineering Division, Winter Annual Meeting in Anaheim, CA, USA, Vol. 59, pp.127-145.
[35] Jouaneh, M., Wang, Z.-X., Dornfeld, D., 1988, " Tracking of Sharp Corners Using a Robot and a Table Manipulator", Proceedings of the USA - Japan Symposium on Flexible Automation, pp. 271-278.
[36] Juttler, B., 1994, "Visualization of Moving Objects Using Dual Quaternion Curves", Computers \& Graphics , Vol. 18, No.3, pp. 315-326.
[37] Kim, J.-H., Ryuh, B.-S., Pennock, G.R., 2001, "Development of a Trajectory Generation Method for a Five-axis NC Machine", Journal of Mechanism and Machine Theory, Vol. 36, No.8, pp. 983-996.
[38] Koninckx, B., Van Brussel, H., 2002, "Real-time NURBS interpolator for distributed motion control", Annals of CIRP, Vol. 51, No.1, pp. 315-318.
[39] Koch, P. E., Wang, K., 1988, "Introduction of B-splines to Trajectory Planning for Robot Manipulators", Modeling, Identification and Control, Vol. 9, No.2, pp. 69-80.
[40] Koren, Y. Lin, R. S.,1994, "Real - Time Five Axis Interpolator for Machining Ruled Surfaces", Proceedings of the 1994 International Mechanical Engineering Congress and Exposition, ASME Dynamic System and Control Division, Vol. 55-2, pp. 951959.
[41] Koren, Y.,1995, "Five Axis Surface Interpolators", Annals of CIRP, Vol. 44, No. 1, pp. 379-382.
[42] Koren, Y., Lo, C.C., Shpitalni, M., 1993, "CNC interpolators: Algorithms and Analysis", ASME Production Engineering Division, Proceedings of the 1993 ASME Winter Annual Meeting, Vol. 64, pp. 83-92.
[43] Kruth, J.-P., Kerstens, A., 1998, "Reverse Engineering. Modeling of Free-Form Surfaces From Point Clouds Subject to Boundary Conditions", Journal of Materials Processing Technology, Vol. 76, No. 1-3, pp. 120-127
[44] Kyriakopoulos, K., Saridis, J., George, N., 1988, "Minimum Jerk Path Generation", Proceedings of the IEEE International Conference on Robotics and Automation, Philedelphia, PA, USA, pp.364-369.
[45] Kyriakopoulos, K., Saridis, J., George, N., 1994, "Minimum Jerk for Trajectory Planning and Control", Robotica, Vol. 12, No.2, pp.109-113.
[46] Lai, J.-Y., Chen, H.-S., Lai, W.-B., 1995, "NC Cutting-path Generation From LaserScan Measurement Data", Journal of the Chinese Society of Mechanical Engineers, Vol. 16, No. 3, pp. 245-255.
[47] Langeron, J. M., Duc, E., Lartigue, C., Bourdet, P., 2004, "A New Format for 5-Axis Toolpath Computation Using B-spline Curves", CAD Computer Aided Design, Vol. 26, No. 12, pp. 1219-1229.
[48] Lartigue, C., Duc, E., Affouard, A., 2003, "Tool Path Deformation In 5-Axis Flank Milling Using Envelope Surface", Vol. 35, No.4, pp.375-382.
[49] Lartigue, C., Thiebaut, F., Maekawa, T., 2001, "CNC Tool Path In Terms of B-Spline Curves"", Computer Aided Design, Vol. 33, No. 4, pp. 307-319.
[50] 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, Vol. 40, No. 10, pp. 1513-1526.
[51] Lorenzo, S., Bruno, S., 1996, Modeling and Control of Robot Manipulators, McGrawHill Companies, New York.
[52] Macfarlane, S., Croft, E. A., 2003, "Jerk-Bounded Manipulator Trajectory Planning: Design For Real-Time Applications", Transactions of IEEE on Robotics and Automation, Vol. 19, No.1, pp.42-52.
[53] Makino, H., Ohde, T., 1991, "Motion Control of the Direct Drive Actuators", Annals of CIRP, Vol. 40, pp. 375-378.
[54] Matlab Reference Guide, 1992, The Math Works Inc.
[55] Muller, M., Erdos, G., Xirouchakis, P., 2004, "High Accuracy Spline Interpolation for 5-Axis Machining", Computer Aided Design, Vol. 36, No. 13, pp. 1379-1393.
[56] Park, H., Kim, K., Lee, S.-C., 2000, "Method for Approximate NURBS Curve Compatibility Based On Multiple Curve Refitting", Computer Aided Design, Vol. 32, No. 4, pp.237-252.
[57] Piazzi, A., Visioli, A., 2000, "Global Minimum-Jerk Trajectory Planning of Robot Manipulators", IEEE Transactions on Industrial Electronics, Vol. 47, No. 1, pp.140149.
[58] Piegl, L., Tiller, W., 1997, The Nurbs Book, Springer Verlag.
[59] 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.
[60] Pritshow, G., Rogers, G., 2004, "Decentralized CNC-Interpolation Using Intelligent Drives", University of Stuttgart, Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW).
[61] Qiu, H., Cheng, K., Li, Y., 1997, "Optimal Circular Arc Interpolation for NC Tool Path Generation in Curve Contour Manufacturing", Computer Aided Design, Vol. 29, No. 11, pp.751-760.
[62] Rao, N., Ismail, F., Bedi, S., 1997, "Tool Path Planning for Five-Axis Machining Using The Principal Axis Method", International Journal of Machine Tools and Manufacture, Vol. 37, No. 7, pp.1025-1040.
[63] Rehsteiner, F., Renker, H. J., 1993, "Collision-Free Five-Axis Milling of Twisted Ruled Surfaces", Annals of CIRP, Vol. 42, No.1, pp457-461.
[64] Ruegg, A., Gygax, P., 1992, "Generalized Kinematics Model for three to Five-Axis Milling Machines and their Implementation in a CNC", Annals of CIRP, Vol. 41, No. 1, pp.547-550.
[65] Sarkar, B., Menq, C.-H., 1991, "Smooth-Surface Approximation and Reverse Engineering", Computer Aided Design, Vol. 23, No. 9, pp.623-628.
[66] Sarkar, B., Menq, C.-H., 1991, "Parameter Optimization In Approximating Curves and Surfaces to Measurement Data", Computer Aided Geometric Design, Vol. 8, No.4, pp.267-290.
[67] Shpitalni, M., Koren, Y., Lo, C.C. , 1994, "Real-Time Curve Interpolators", CAD Computer Aided Design, Vol. 26, No. 11, pp. 832-838.
[68] Simon, D., Isik, C., 1993, "Suboptimal Robot Joint Interpolation Within UserSpecified Knot Tolerances", Journal of Robotic Systems, Vol. 10, No.7, pp.889-911.
[69] Siemens AG, 2004, "Sinumerik 840D/840Di/810D Werkzeug und Formenbau Handbuch", Erlangen, Germany.
[70] Siemens AG, 2002, "Advanced Programming Guide SINUMERIK 840D/840Di/810D", Erlangen, Germany.
[71] Srinivasan, L.N., Ge, Q.J., 1998, " Fine Tuning of Rational B-Spline Motions", ASME Journal of Mechanical Design, Vol. 120, No. 1, pp.46-51.
[72] Srinivasan, L.N., Ge, Q.J., 1996, "Parametric Continuous and Smooth Motion Interpolation", ASME Journal of Mechanical Design, Vol. 118, No. 4, pp.494-498
[73] Trepanier, J.-Y., Lepine, J.L., Pepin, F., 2000, " Optimized Geometric Representation for Wing Profiles Using NURBS", Journal of Canadian Aeronautics and Space, Vol. 46, No.1, pp.12-19.
[74] Tsai, Y.-F., Farouki, R.T., Feldman, B., 2001, "Performance Analysis of CNC Interpolators for Time-Dependent Feedrates Along PH Curves", Computer Aided Geometric Design, Vol. 18, No. 3, pp.245-265.
[75] Vassilev, T., I., 1996, "Fair Interpolation and Approximation of B-splines by Energy Minimization and Points Insertion", Computer Aided Design, Vol. 28, No. 9, pp. 753760.
[76] Visioli, A., 2000, "Trajectory Planning of Robot Manipulators By Using Algebraic and Trigonometric Splines", Robotica, Vol. 18, No. 6, pp. 611-631.
[77] Voliotis, S.D., 1992, "Orientation Planning in Continuous Path Applications for Wrist Partitioned Manipulators", IEE Proceedings, Part D: Control Theory and Applications, Vol. 139, No. 6, pp.495-502.
[78] Wang, F-C., Yang, D.C.H., 1993, "Nearly Arc-Length Parameterized Quintic-Spline Interpolation for Precision Machining", Computer Aided Design, Vol. 25, No. 5, pp. 281-288.
[79] Wang, F.-C., Schofield, S., Wright, P.K., 1996, "Real Time Quintic Spline Interpolator For An Open Architecture Machine Tool", ASME Dynamic Systems and Control Division, Proceedings of the 1996 ASME International Mechanical Engineering Congress and Exposition, Vol. 58, pp. 291-297.
[80] Wang, F.-C., Wright, P.K., 1998, "Open Architecture Controllers for Machine Tools, Part 2: A Real Time Quintic Spline Interpolator", ASME Journal of Manufacturing Science and Engineering, Vol. 120, No. 2, pp. 425-432.
[81] Weck, M., Meylahn, A., Hardebush, C., 1999, "Innovative Algorithms for SplineBased CNC Controller", Production Engineering Research and Development in Germany: Annals of the German Academic Society for Production Engineering, Vol. 6, No. 1, pp.83-86.
[82] Wu, C. H., Jou, C.C., 1989, "Planning and Control of Robot Orientational Path", IEEE Transactions on Systems, Manufacturing and Cybernetics, Vol. 19, No. 5, pp.12341241.
[83] Xu, H.-Y., Dai, J., Tam, H.-Y., Zhou, Y., 2003, "Angular Feedrate Interpolation for Three-Dimensional Implicit Curves", International Journal of Production Research, Vol. 41, No.15, pp.3461-3478.
[84] Yang, D.C.H., Golub, A.D., 1992, "Improved Trajectory Generation for Precision CNC Milling of Arbitrary Contours and Surfaces", ASME Production Engineering Division, Winter Annual Meeting, Vol. 52, pp. 105-117.
[85] Yang, D.C.H., Golub, Alex D., 1994, "Precision Trajectory Generation For CNC Milling of Arbitrary Contours and Surfaces", International Journal of Machine Tools \& Manufacture, Vol. 34, No. 7, pp. 1005-1018.
[86] Yang, D.C.H., Kong, T., 1994, "Parametric Interpolator Versus Linear Interpolator For Precision CNC Machining", Vol.26, No. 3, pp.225-234.
[87] Yang, M.-Y., Park, J.-H., 2002, "A Study On An Open Architecture Cnc System With A Nurbs Interpolator For WEDM", International Journal of Advanced Manufacturing Technology, Vol. 19, No. 9, pp. 664-6683.
[88] Yau, H.-T., Chen, J.-S., 1997, "Reverse Engineering of Complex Geometry Using Rational B-Splines", International Journal of Advanced Manufacturing Technology, Vol. 13, No. 8, pp. 548-555.
[89] Yau, H.-T., Kuo, M.-J., 2001, "NURBS Machining and Feed Rate Adjustment For High-Speed Cutting of Complex Sculptured Surfaces", International Journal of Production Research, Vol. 39, No. 1, pp. 21-41.
[90] Yeh, S.-S., Hsu, P.-L., 2002, "Adaptive-Feedrate Interpolation For Parametric Curves With A Confined Chord Error", Computer Aided Design, Vol. 34, No. 3, pp. 229-237
[91] Yeung, C.-H., 2004, M.A.Sc. Thesis: "A Three - Axis Virtual Computer NumericalControlled (CNC) System", University of British Columbia, Department of Mechanical Engineering, BC, Canada.
[92] Zhang, Q.G., Greenway, R.B., 1998, "Development and Implementation of A NURBS Curve Motion Interpolator", Robotics and Computer-Integrated Manufacturing, Vol. 14, No.1, pp. 27-36.
[93] Zhiming, X., Jincheng, C., Zhengjin, F., 2002, "Performance Evaluation of a RealTime Interpolation Algorithm for NURBS Curves", International Journal of Advanced Manufacturing Technology, Vol. 20, No.4, pp.270-276.

## Appendix A

## Formulation of the Minimum Jerk Feed Profile

The displacement profile for the $\mathrm{k}^{\text {th }}$ knot span segment was expressed in Chapter 4 as:

$$
\begin{equation*}
s_{k}(\tau)=A_{k} \tau^{5}+B_{k} \tau^{4}+C_{k} \tau^{3}+D_{k} \tau^{2}+E_{k} \tau+F_{k}, 0 \leq \tau \leq T_{k} \tag{A.1}
\end{equation*}
$$

In order to avoid any ill conditioning, the time parameter $\tau$ and segment durations can be normalized by the longest expected segment duration $T_{\max }$ at the beginning, resulting in the following feed spline equation,

$$
\begin{equation*}
s_{k}(v)=a_{k} \nu^{5}+b_{k} \nu^{4}+c_{k} \nu^{3}+d_{k} v^{2}+e_{k} v+f_{k} \quad, \quad 0 \leq v \leq \lambda_{k} \tag{A.2}
\end{equation*}
$$

where $\quad v=\tau / T_{\max }, \lambda_{k}=T_{k} / T_{\max } \quad$ and $\quad a_{k}=T_{\max }^{5} A_{k}, b_{k}=T_{\max }^{4} B_{k}, \ldots, f_{k}=F_{k} . \quad$ The normalized velocity, acceleration, and jerk equations of the given quintic time spline are as follows:

$$
\left.\begin{array}{l}
\dot{s}_{k}(v)=\left(1 / T_{\max }\right)\left[5 a_{k} v^{4}+4 b_{k} v^{3}+3 c_{k} v^{2}+2 d_{k} v+e_{k}\right] \\
\dddot{s}_{k}(v)=\left(1 / T_{\max }^{2}\right)\left[20 a_{k} v^{3}+12 b_{k} v^{2}+6 c_{k} v+2 d_{k}\right]  \tag{A.3}\\
\dddot{s}_{k}(v)=\left(1 / T_{\max }^{3}\right)\left[60 a_{k} v^{2}+24 b_{k} v+6 c_{k}\right]
\end{array}\right\} .
$$

The local minimum jerk objective function was given in Eq. (4.35) (Chapter 4) is expressed using the normalized time spline parameters as:

$$
J_{V k}=\frac{1}{T_{\max }^{5}} \int_{0}^{\lambda_{k}}\left[\begin{array}{l}
3600 a_{k}^{2} v^{4}+2880 a_{k} b_{k} v^{3}+  \tag{A.4}\\
\left(720 a_{k} c_{k}+576 b_{k}^{2}\right) v^{2}+288 b_{k} c_{k} v+36 c_{k}^{2}
\end{array}\right] d v .
$$

The integration in Eq.(A.4) results in the following matrix formulation:
where $\theta_{k}=\left[\begin{array}{llll}a_{k} & b_{k} & \ldots & f_{k}\end{array}\right]^{T}$ is the normalized parameters of the displacement spline. By dropping the $1 / T_{\text {max }}^{5}$ at the beginning of Eq.(A.5) and replacing with $1 / 2$ for mathematical convenience, the minimum jerk objective for the $\mathrm{k}^{\text {th }}$ feed segment can be rewritten as:

$$
\begin{equation*}
J_{V k}=\frac{1}{2} \theta_{k}^{T} \cdot K_{k}\left(\lambda_{k}\right) \cdot \theta_{k}=\frac{1}{2} \theta_{k}^{T} \cdot K_{k} \cdot \theta_{k} . \tag{A.6}
\end{equation*}
$$

The overall minimum jerk objective in Eq.(A.6) can then be stacked for all the segments, where the local parameters and weighting matrices for local segments are $\theta_{k}$ and $K_{k}$ as:

$$
J_{V}=\sum_{k=0}^{N_{s-1}} J_{k}=\frac{1}{2} \cdot \underbrace{\left[\begin{array}{llll}
\theta_{0}^{T} & \theta_{1}^{T} & \cdots & \theta_{N s-1}^{T}
\end{array}\right] \cdot \underbrace{\left[\begin{array}{cccc}
K_{0} & & & 0  \tag{A.7}\\
& K_{1} & & \\
& & \ddots & \\
0 & & & K_{N s-1}
\end{array}\right]}_{K(\lambda)} \cdot \underbrace{\left[\begin{array}{c}
\theta_{0} \\
\theta_{1} \\
\vdots \\
\theta_{N_{s-1}}
\end{array}\right]}_{\theta_{6 N s \times 1}} . . . . ~ . . ~}_{\theta_{1 \times 0 N s}}
$$

Position $\left(\mathrm{C}^{0}\right), \mathrm{C}^{1}, \mathrm{C}^{2}$ and $\mathrm{C}^{3}$ boundary conditions are imposed at the segment junctions to connect the time spline segments together over the complete toolpath. For the $\mathrm{k}^{\text {th }}$ segment, the following position boundary conditions must hold,

$$
\begin{equation*}
s_{k}(0)=\sum_{i=0}^{k-1} S_{i}, \quad s_{k}\left(\lambda_{k}\right)=\sum_{i=0}^{k} S_{i} \tag{A.8}
\end{equation*}
$$

where $S_{k}$ is the arc length of the $\mathrm{k}^{\text {th }}$ segment. The boundary condition is rewritten in the matrix form as:

$$
\underbrace{\left[\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 1  \tag{A.9}\\
\lambda_{k}^{5} & \lambda_{k}^{4} & \lambda_{k}^{3} & \lambda_{k}^{2} & \lambda_{k} & 1
\end{array}\right]}_{L_{k}^{0}\left(\lambda_{k}\right)} \cdot \underbrace{\left[\begin{array}{c}
a_{k} \\
b_{k} \\
\vdots \\
f_{k}
\end{array}\right]}_{\theta_{k}}=\underbrace{\left[\begin{array}{c}
\sum_{i=1}^{k-1} S_{i} \\
\sum_{i=1}^{k} S_{i}
\end{array}\right]}_{\xi_{k}^{0}}
$$

The knot position constraints from all segments can be grouped together as:

$$
\underbrace{\left[\begin{array}{cccc}
L_{0}^{0} & & & 0  \tag{A.10}\\
& L_{1}^{0} & & \\
& & \ddots & \\
0 & & & L_{N s-1}^{0}
\end{array}\right]}_{L^{0}(\lambda)} \cdot \underbrace{\left[\begin{array}{c}
\theta_{0} \\
\theta_{1} \\
\vdots \\
\theta_{N s-1}
\end{array}\right]}_{\theta}=\underbrace{\left[\begin{array}{c}
\xi_{0}^{0} \\
\xi_{1}^{0} \\
\vdots \\
\xi_{N s-1}^{0}
\end{array}\right] . .}_{\xi^{0}}
$$

Following that, the jerk continuous feed profile is generated by imposing the velocity, acceleration, and jerk continuity constraints. For instance, at the junction of $k^{\text {th }}$ and $k+1^{\text {st }}$ segments the boundary conditions are given in Eq.(4.32) (Chapter 4) as,

$$
\left.\left.\begin{array}{l}
\dot{s}_{k}\left(\lambda_{k}\right)=\dot{s}_{k+1}(0) \\
\dddot{\mathrm{s}}_{k}\left(\lambda_{k}\right)=\dddot{\mathrm{s}}_{k+1}(0)  \tag{A.11}\\
\dddot{\mathrm{s}}_{k}\left(\lambda_{k}\right)=\dddot{\mathrm{s}}_{k+1}(0)
\end{array}\right\} \Rightarrow \begin{array}{l}
\dot{s}_{k}\left(\lambda_{k}\right)-\dot{s}_{k+1}(0)=0 \\
\ddot{\mathrm{~s}}_{k}\left(\lambda_{k}\right)-\dddot{\mathrm{s}}_{k+1}(0)=0 \\
\dddot{s}_{k}\left(\lambda_{k}\right)-\dddot{\mathrm{s}}_{k+1}(0)=0
\end{array}\right\}
$$

where $k=0,1, \ldots, N_{s}-2$. Substituting the expressions for for $\dot{s}_{k}, \ddot{s}_{k}$ and $\dddot{s}_{k}$, from Eq. (A.3) into Eq. (A.11), the matrix equation is written,

$$
\underbrace{\left[\begin{array}{ccccccccccccc}
5 \lambda_{k}^{4} & 4 \lambda_{k}^{3} & 3 \lambda_{k}^{2} & 2 \lambda_{k} & 1 & 0 & \vdots & 0 & 0 & 0 & 0 & -1 & 0 \\
20 \lambda_{k}^{3} & 12 \lambda_{k}^{2} & 6 \lambda_{k} & 2 & 0 & 0 & \vdots & 0 & 0 & 0 & -2 & 0 & 0 \\
60 \lambda_{k}^{2} & 24 \lambda_{k} & 6 & 0 & 0 & 0 & \vdots & 0 & 0 & -6 & 0 & 0 & 0
\end{array}\right]}_{L_{k}^{133}\left(\lambda_{k}\right)} \cdot\left[\begin{array}{c}
\theta_{k} \\
\cdots \\
\theta_{k+1}
\end{array}\right]=\left[\begin{array}{l}
0 \\
0 \\
0
\end{array}\right]
$$

The derivative boundary conditions for $N_{s}-1$ segment junctions (Eq. (A.12)) are grouped together and stacked in the following matrix equation as:

$$
\underbrace{\left[\begin{array}{llll}
L_{0}^{123} & 0_{3 \times 6} & & 0  \tag{A.13}\\
0_{3 \times 6} & L_{1}^{123} & & \\
& & \ddots & \\
0 & & & L_{N s-2}^{123}
\end{array}\right]}_{L^{123}(\lambda)} \cdot \underbrace{\left[\begin{array}{c}
\theta_{0} \\
\theta_{1} \\
\vdots \\
\theta_{N s-1}
\end{array}\right]}_{\theta}=\underbrace{0_{3(N s-1) \times 1}}_{\xi^{123}}
$$

If initial conditions of velocity $\left(\dot{s}_{\text {init }}\right)$, acceleration $\left(\dddot{s}_{\text {init }}\right)$, and jerk ( $\dddot{s}_{\text {init }}$ ) are provided, they can be included into the formulation as, $\dot{\mathrm{s}}_{0}(0)=\dot{\mathrm{s}}_{\text {init }}, \ddot{\mathrm{s}}_{0}(0)=\dddot{s}_{\text {init }}$ and $\dddot{\varsigma}_{0}(0)=\dddot{\mathrm{s}}_{\text {init }}$ in the feed profile. The boundary conditions expressed using the Eq. (A.3) are given in the following matrix equation as:

$$
\underbrace{\left[\begin{array}{llllllll}
0 & 0 & 0 & 0 & 1 & 0 & \vdots &  \tag{A.14}\\
0 & 0 & 0 & 2 & 0 & 0 & \vdots & 0_{3 \times 6(N s-1)} \\
0 & 0 & 6 & 0 & 0 & 0 & \vdots &
\end{array}\right]}_{L_{L_{i n i t}}} \cdot \underbrace{\left[\begin{array}{c}
\theta_{0} \\
\theta_{1} \\
\vdots \\
\theta_{N s-1}
\end{array}\right]}_{\theta}=\underbrace{\left[\begin{array}{c}
T_{\max } \dot{s}_{0} \\
T_{\max }^{2} \ddot{s}_{0} \\
T_{\max }^{3} \ddot{z}_{0}
\end{array}\right]}_{\xi_{\text {mini }}} .
$$

Similarly, if final kinematic conditions of velocity ( $\dot{s}_{\text {final }}$ ), acceleration ( $\ddot{s}_{\text {final }}$ ), and jerk $\left(\dddot{s}_{f \text { fnal }}\right)$ are provided at the end of the toolpath, they can be included as $\dot{s}_{N s-1}\left(\lambda_{N s-1}\right)=\dot{s}_{\text {final }}$, $\dddot{s}_{N s-1}\left(\lambda_{N s-1}\right)=\dddot{s}_{f n a l}, \dddot{s}_{N s-1}\left(\lambda_{N s-1}\right)=\dddot{s}_{f i n a l}$ and written in the matrix form:

At the segment junctions; position, $1^{\text {st }}, 2^{\text {nd }}$, and $3^{\text {rd }}$ derivative continuity constraints are given in Eq.(A.9) and Eq.(A.12). The initial and final boundary conditions are provided for the feed profile from Eq. (A.14) and Eq. (A.15). All of the constraints are stacked together in the following matrix equation form:

$$
\underbrace{\left[\begin{array}{l}
l^{0}(\lambda)  \tag{A.16}\\
l^{123}(\lambda) \\
L_{\text {init }} \\
L_{\text {final }}\left(\lambda_{N s-1}\right)
\end{array}\right]}_{L(\lambda)} \cdot \theta=\underbrace{\left[\begin{array}{l}
\xi^{0} \\
\xi^{123} \\
\xi_{\text {init }} \\
\xi_{\text {final }}
\end{array}\right]}_{\xi} \Rightarrow L(\lambda) \cdot \theta=\xi
$$

The minimum jerk optimization problem can now be expressed using Eq. (A.7) and Eq. (A.16) as:

$$
\begin{equation*}
\min _{\theta} \frac{1}{2} \theta^{T} \cdot K(\lambda) \cdot \theta \quad \text { subject to : } L(\lambda) \cdot \theta=\xi \tag{A.17}
\end{equation*}
$$

The above constrained quadratic minimization problem can then be solved using the Lagrange Multipliers method. Introducing the vector of Lagrange multipliers $\Lambda=\left[\begin{array}{llll}\Lambda_{0} & \Lambda_{1} & \ldots & \Lambda_{5 N s+2}\end{array}\right]^{T}$, the objective function can be expressed as,

$$
\begin{equation*}
\bar{J}_{V}(\theta, \Lambda)=\frac{1}{2} \theta^{T} \cdot K(\lambda) \cdot \theta+\grave{\Lambda}^{T}[L(\lambda) \cdot \theta-\xi] \tag{A.18}
\end{equation*}
$$

By equating the partial derivatives to zero ( $\partial \bar{J}_{V} / \partial \theta=0$ and $\partial \bar{J}_{V} / \partial \Lambda=0$ ) yields the linear equation system:

$$
\left.\begin{array}{l}
K(\lambda) \cdot \theta+L^{T}(\lambda) \cdot \Lambda=0  \tag{A.19}\\
L(\lambda) \cdot \theta=\xi
\end{array}\right\} \Rightarrow\left[\begin{array}{cc}
K(\lambda) & L^{T}(\lambda) \\
L(\lambda) & 0
\end{array}\right] \cdot\left[\begin{array}{l}
\theta \\
\Lambda
\end{array}\right]=\left[\begin{array}{l}
0 \\
\xi
\end{array}\right]
$$

which has full rank as long as all the segment durations are nonzero. The solution of the above matrix equation yields the coefficients $(\theta)$ of the minimum jerk feed profile. The coefficients are de-normalized as $A_{k}=1 / T_{\max }^{s} \cdot a_{k}, B_{k}=1 / T_{\max }^{4} \cdot b_{k}, \ldots, F_{k}=f_{k}$ and the feed profile is generated using Eq.(A.3).

## Appendix B

## Computation of the Gradients in Feedrate Optimization

## B.1. Gradient of the Objective

The minimum time objective was given in the Eq. (4.36) in Chapter 4 as:

$$
T_{\Sigma}=\sum_{k=0}^{N_{s-1}} T_{k},
$$

and in normalized form of

$$
\begin{equation*}
T_{\Sigma}=\sum_{k=0}^{N_{s}-1} T_{k}=T_{\max } \sum_{k=0}^{N_{s}-1} \lambda_{k} \tag{B.1}
\end{equation*}
$$

Hence, its gradient with respect to the normalized time duration of the $\mathrm{k}^{\text {th }}$ segment in the B-spline representation $\lambda_{k}$ is expressed as:

$$
\begin{equation*}
\frac{\partial T_{\mathcal{E}}}{\partial \lambda^{T}}=T_{\max } \tag{B.2}
\end{equation*}
$$

The gradient with respect to the vector of all normalized segment durations $\lambda=\left[\begin{array}{llll}\lambda_{0} & \lambda_{1} & \ldots & \lambda_{N s-1}\end{array}\right]^{T}$ is obtained in a vector form as:

$$
\frac{\partial T_{\Sigma}}{\partial \lambda^{T}}=T_{\max } \cdot\left[\begin{array}{llll}
1 & 1 & \ldots & 1 \tag{B.3}
\end{array}\right]_{1 \times N s}
$$

## B.2. Gradient of the Feed Profile Coefficients

The vector of the Quintic displacement profile coefficients: $\theta$ is obtained by solving Eq. (A.19) in Appendix A as:

$$
\left.\begin{array}{l}
K(\lambda) \cdot \theta+L^{T}(\lambda) \cdot \Lambda=0  \tag{B.4}\\
L(\lambda) \cdot \theta=\xi
\end{array}\right\} \Rightarrow \underbrace{\left[\begin{array}{cc}
K(\lambda) & L^{T}(\lambda) \\
L(\lambda) & 0
\end{array}\right]}_{A(\lambda)} \cdot \underbrace{\left[\begin{array}{l}
\theta \\
\Lambda
\end{array}\right]}_{X}=\underbrace{\left[\begin{array}{l}
0 \\
\xi
\end{array}\right] .}_{b}
$$

It is known that the $\mathbf{b}$ vector on the right side of Eq. (B.4) is a constant depending only on the toolpath arc lengths and the given boundary conditions. The differentiation of both sides of the Eq.(B.4) with respect to the design variables $\lambda_{j}$ yields the following equation:

$$
\begin{equation*}
\frac{\partial A}{\partial \lambda_{j}} \cdot X+A \cdot \frac{\partial X}{\partial \lambda_{j}}=0 \Rightarrow A \frac{\partial X}{\partial \lambda_{j}}=-\frac{\partial A}{\partial \lambda_{j}} \cdot X \tag{B.5}
\end{equation*}
$$

A and $X$ are obtained from the solution of Eq.(B.4), the gradient expression $\partial A / \partial \lambda_{j}$ is constructed by considering the definitions of the $K(\lambda)$ and $L(\lambda)$ matrices given in Eq. (A.7) and (A.19) (Appendix A). Basically, the expression for $\partial X / \partial \lambda_{j}$ is obtained by solving the linear equation system in Eq.(B.5), and the gradient of the feed profile coefficients $\partial \theta / \partial \lambda_{j}$ is extracted from:

$$
\frac{\partial X}{\partial \lambda_{j}}=\left[\begin{array}{ll}
\partial \theta^{T} / \partial \lambda_{j} & \partial \Delta^{T} / \partial \lambda_{j} \tag{B.6}
\end{array}\right]^{T}
$$

The gradient matrix, consisting of partial derivatives of the feed profile coefficients with respect to all normalized segment durations is constructed as:

$$
\frac{\partial \theta}{\partial \lambda^{T}}=\left[\begin{array}{cccc}
\partial a_{0} / \partial \lambda_{0} & \partial a_{0} / \partial \lambda_{1} & \ldots & \partial a_{0} / \partial \lambda_{N a-1}  \tag{B.7}\\
\partial b_{0} / \partial \lambda_{0} & \partial b_{0} / \partial \lambda_{1} & \ldots & \partial b_{0} / \partial \lambda_{N a-1} \\
\vdots & \vdots & \ldots & \vdots \\
\partial f_{0} / \partial \lambda_{0} & \partial f_{0} / \partial \lambda_{1} & \ldots & \partial f_{0} / \partial \lambda_{N a-1} \\
\partial a_{1} / \partial \lambda_{0} & \partial a_{1} / \partial \lambda_{1} & \ldots & \partial a_{1} / \partial \lambda_{N a-1} \\
\partial b_{1} / \partial \lambda_{0} & \partial b_{1} / \partial \lambda_{1} & \ldots & \partial b_{1} / \partial \lambda_{N a-1} \\
\vdots & \vdots & \ldots & \vdots \\
\partial f_{1} / \partial \lambda_{0} & \partial f_{1} / \partial \lambda_{1} & \ldots & \partial f_{1} / \partial \lambda_{N a-1} \\
& & & \ldots \\
\vdots & \vdots & \ddots & \ldots \\
& & & \ldots \\
\partial a_{N s-1} / \partial \lambda_{0} & \partial a_{N s-1} / \partial \lambda_{1} & \ldots & \partial a_{N s-1} / \partial \lambda_{N s-1} \\
\partial b_{N s-1} / \partial \lambda_{0} & \partial b_{N s-1} / \partial \lambda_{1} & \ldots & \partial b_{N s-1} / \partial \lambda_{N s-1} \\
\vdots & \vdots & \ldots & \vdots \\
\partial f_{N s-1} / \partial \lambda_{0} & \partial f_{N s-1} / \partial \lambda_{1} & \ldots & \partial f_{N s-1} / \partial \lambda_{N s-1}
\end{array}\right]_{6 N s \times N s}
$$

## B.3. Gradient of the Kinematic Profiles

The arc displacement, feedrate, tangential acceleration, and jerk profiles in the $\mathrm{k}^{\text {th }}$ segment have been given as:

$$
\left.\begin{array}{l}
s_{k}(v)=a_{k} v^{5}+b_{k} v^{4}+c_{k} v^{3}+d_{k} v^{2}+e_{k} v+f_{k} \\
\dot{s}_{k}(v)=\left(1 / T_{\max }\right)\left(5 a_{k} v^{4}+4 b_{k} v^{3}+3 c_{k} v^{2}+2 d_{k} v+e_{k}\right) \\
\ddot{s}_{k}(v)=\left(1 / T_{\max }^{2}\right)\left(20 a_{k} v^{3}+12 b_{k} v^{2}+6 c_{k} v+2 d_{k}\right)  \tag{B.8}\\
\dddot{s}_{k}(v)=\left(1 / T_{\max }^{3}\right)\left(60 a_{k} v^{2}+24 b_{k} v+6 c_{k}\right)
\end{array}\right\}
$$

where $0 \leq v \leq \lambda_{k}$. The normalized time parameter $v$ is redefined as:

The displacement profile is then written as:

$$
\begin{equation*}
s_{k}\left(\lambda_{k} \mu\right)=a_{k} \lambda_{k}^{5} \mu^{5}+b_{k} \lambda_{k}^{4} \mu^{4}+c_{k} \lambda_{k}^{3} \mu^{3}+d_{k} \lambda_{k}^{2} \mu^{2}+e_{k} \lambda_{k} \mu+f_{k} \tag{B.10}
\end{equation*}
$$

The gradient of the displacement profile in the $\mathrm{k}^{\text {th }}$ segment with respect to the vector of normalized segment durations $\lambda=\left[\begin{array}{llll}\lambda_{0} & \lambda_{1} & \ldots & \lambda_{N s-1}\end{array}\right]$ is obtained as:

$$
\begin{align*}
& \frac{\partial s_{k}}{\partial \lambda^{T}}=\lambda_{k}^{5} \mu^{5} \frac{\partial a_{k}}{\partial \lambda^{T}}+\lambda_{k}^{4} \mu^{4} \frac{\partial b_{k}}{\partial \lambda^{T}}+\lambda_{k}^{3} \mu^{3} \frac{\partial c_{k}}{\partial \lambda^{T}}+\lambda_{k}^{2} \mu^{2} \frac{\partial d_{k}}{\partial \lambda^{T}}+\lambda_{k} \mu \frac{\partial e_{k}}{\partial \lambda^{T}}+\frac{\partial f_{k}}{\partial \lambda^{T}}+\ldots  \tag{B.11}\\
& {\left[5 a_{k} \mu^{5} \lambda_{k}^{4}+4 b_{k} \mu^{4} \lambda_{k}^{3}+3 c_{k} \mu^{3} \lambda_{k}^{2}+2 d_{k} \mu^{2} \lambda_{k}+e_{k} \mu\right] \cdot \frac{\partial \lambda_{k}}{\partial \lambda^{T}}}
\end{align*}
$$

where the gradients $\partial a_{k} / \partial \lambda^{T}, \partial b_{k} / \partial \lambda^{T}, \ldots, \partial f_{k} / \partial \lambda^{T}$ are obtained form Eq.(B.7). The gradient of the $\mathrm{k}^{\text {th }}$ normalized segment duration with respect to all normalized segment durations is $\partial \lambda_{k} / \partial \lambda^{T}=\left[\begin{array}{lllllll}0 & \ldots & 0 & 1 & 0 & \ldots & 0\end{array}\right]$ where only the $\mathrm{k}^{\text {th }}$ entry is one. Considering the definition of $\dot{s}$ in terms of normalized parameter in Eq.(B.8) and factoring out the $\mu$ term out of the expression in Eq.(B.11), the gradient of the displacement profile with respect to the normalized segment duration is obtained as:

$$
\begin{equation*}
\frac{\partial s_{k}}{\partial \lambda^{T}}=v^{5} \frac{\partial a_{k}}{\partial \lambda^{T}}+v^{4} \frac{\partial b_{k}}{\partial \lambda^{T}}+v^{3} \frac{\partial c_{k}}{\partial \lambda^{T}}+v^{2} \frac{\partial d_{k}}{\partial \lambda^{T}}+v \frac{\partial e_{k}}{\partial \lambda^{T}}+\frac{\partial f_{k}}{\partial \lambda^{T}}+\mu T_{\max } \dot{s}_{k}(v) \frac{\partial \lambda_{k}}{\partial \lambda^{T}} \tag{B.12}
\end{equation*}
$$

Following a similar procedure, the gradients of the feedrate, tangential acceleration, and jerk profiles are also obtained as:

$$
\left.\begin{array}{l}
\frac{\partial \dot{s}_{k}}{\partial \lambda^{T}}=\left[5 v^{4} \frac{\partial a_{k}}{\partial \lambda^{T}}+4 v^{3} \frac{\partial b_{k}}{\partial \lambda^{T}}+3 v^{2} \frac{\partial c_{k}}{\partial \lambda^{T}}+2 v \frac{\partial d_{k}}{\partial \lambda^{T}}+\frac{\partial e_{k}}{\partial \lambda^{T}}+\mu T_{\max }^{2} \ddot{s}_{k}(v) \frac{\partial \lambda_{k}}{\partial \lambda^{T}}\right] \\
\frac{\partial \ddot{s}_{k}}{\partial \lambda^{T}}=\left[20 v^{3} \frac{\partial a_{k}}{\partial \lambda^{T}}+12 v^{2} \frac{\partial b_{k}}{\partial \lambda^{T}}+6 v \frac{\partial c_{k}}{\partial \lambda^{T}}+2 \frac{\partial d_{k}}{\partial \lambda^{T}}+\mu T_{\max }^{3} \dddot{s}_{k}(v) \frac{\partial \lambda_{k}}{\partial \lambda^{T}}\right]  \tag{B.13}\\
\frac{\partial \ddot{s}_{k}}{\partial \lambda^{T}}=\left[60 v^{2} \frac{\partial a_{k}}{\partial \lambda^{T}}+24 v \frac{\partial b_{k}}{\partial \lambda^{T}}+6 \frac{\partial c_{k}}{\partial \lambda^{T}}+\mu T_{\max }^{4} \cdots \cdots_{k}(v) \frac{\partial \lambda_{k}}{\partial \lambda^{T}}\right]
\end{array}\right\}
$$

## B.4. Gradients of the Constraints

The velocity, acceleration and jerk constraints have been given in Chapter 4, Eq.(4.60) as:

$$
\left.\begin{array}{l}
C_{f}^{+}(t)=\dot{s}(t) / f_{\text {max }}-1 \leq 0, \quad C_{f}^{-}(t)=-\dot{s}(t) / f_{\max } \leq 0 \\
C_{\dot{X}}^{+}(t)=\dot{X}(t) / v_{X \text { max }}-1 \leq 0, C_{\dot{X}}^{-}(t)=-\dot{X}(t) / v_{X \text { max }}-1 \leq 0 \\
C_{\dot{Y}}^{+}(t)=\dot{Y}(t) / v_{Y \text { max }}-1 \leq 0, \quad C_{\dot{Y}}^{-}(t)=-\dot{Y}(t) / v_{Y \text { max }}-1 \leq 0 \\
C_{Z}^{+}(t)=\dot{Z}(t) / v_{Z \max }-1 \leq 0, \quad C_{\dot{Z}}^{-}(t)=-\dot{Z}(t) / v_{Z_{\text {max }}}-1 \leq 0 \\
C_{\ddot{\theta}_{A}}^{+}(t)=\dot{\theta}_{A}(t) / v_{\theta_{A} \max }-1 \leq 0, \quad C_{\dot{\theta}_{A}}^{-}(t)=-\dot{\theta}_{A}(t) / v_{\theta_{A} \max }-1 \leq 0 \\
C_{\theta_{C}}^{+}(t)=\dot{\theta}_{C}(t) / v_{\theta_{C} \max }-1 \leq 0, \quad C_{\dot{\theta}_{C}}^{-}(t)=-\dot{\theta}_{C}(t) / v_{\theta_{C} \max }-1 \leq 0
\end{array}\right\} \text { Velocity, }
$$

The feedrate upper bound constraints $C_{f}^{+}(t)=\dot{s}(t) / f_{\max }-1 \leq 0$, its gradient with respect to normalized segment durations s obtained as:

$$
\begin{equation*}
\frac{\partial C_{f}^{+}}{\partial \lambda^{T}}=\frac{1}{f_{\max }} \frac{\partial \dot{s}(t)}{\partial \lambda^{T}} \tag{B.15}
\end{equation*}
$$

where the gradient of the feedrate profile $\left(\partial \dot{s} / \partial \lambda^{T}\right)$ is given in Eq.(B.13). The axis kinematic profiles have been given in Chapter 4, Eq. (5.58) as:

$$
\left.\begin{array}{l}
\dot{M}(t)=M_{s} \cdot \dot{s}(t) \\
\ddot{M}(t)=M_{s s} \cdot \dot{s}(t)^{2}+M_{s} \cdot \ddot{s}(t)  \tag{B.16}\\
\ddot{M}(t)=M_{s s s} \cdot \dot{s}(t)^{3}+3 M_{s s} \cdot \dot{s}(t) \cdot \ddot{s}(t)+M_{s} \cdot \dddot{s}(t)
\end{array}\right\}
$$

where $M(s)=\left[\begin{array}{lllll}X(s) & Y(s) & Z(s) & \theta_{A}(s) & \theta_{C}(s)\end{array}\right]^{T}$. The X axis velocity upper bound constraint can be obtained by combining Eq.(B.14) and (B.16) as:

$$
\begin{equation*}
C_{\dot{X}}^{+}=\frac{1}{v_{\mathrm{x} \max }} \dot{X}(t)-1=\frac{1}{v_{\mathrm{x} \max }} X_{s}(s) \cdot \dot{s}(t)-1 \leq 0 \tag{B.17}
\end{equation*}
$$

and its gradient with respect to normalized segment durations is obtained as:

$$
\begin{equation*}
\frac{\partial C_{X}^{+}}{\partial \lambda^{T}}=\frac{1}{v_{\mathrm{x} \max }}\left[X_{s s} \dot{s}(t) \frac{\partial s}{\partial \lambda^{T}}+X_{s} \frac{\partial \dot{s}}{\partial \lambda^{T}}\right] \tag{B.18}
\end{equation*}
$$

The $X$ axis acceleration upper bound constraint is obtained from Eq.(B.14) and Eq. (B.16) as:

$$
\begin{equation*}
C_{\ddot{X}}^{+}=\frac{1}{a_{\mathrm{x} \max }} \ddot{X}(t)-1=\frac{1}{a_{\mathrm{x} \max }}\left[X_{s s}(s) \cdot \dot{s}(t)^{2}+X_{s} \cdot \ddot{s}(t)\right]-1 \leq 0 \tag{B.19}
\end{equation*}
$$

and its gradient with respect to normalized segment duration is obtained as:

$$
\begin{equation*}
\frac{\partial C_{\dot{X}}^{+}}{\partial \lambda^{T}}=\frac{1}{a_{\mathrm{x} \max }}\left[\left(X_{s s s} \dot{s}(t)^{2}+X_{s s} \ddot{s}(t)\right) \frac{\partial s}{\partial \lambda^{T}}+2 X_{s s} \dot{s}(t) \frac{\partial \dot{s}}{\partial \lambda^{T}}+X_{s} \frac{\partial \ddot{s}}{\partial \lambda^{T}}\right] \tag{B.20}
\end{equation*}
$$

The X axis jerk upper bound constraint can also be obtained from Eq.(B.14) and Eq. (B.16) as:

$$
\begin{equation*}
C_{\ddot{X}}^{+}=\frac{1}{j_{\mathrm{x} \max }} \dddot{X}(t)-1=\frac{1}{j_{\mathrm{x} \max }}\left[X_{s s s}(s) \cdot \dot{s}(t)^{3}+3 X_{s s} \cdot \dot{s}(t) \cdot \ddot{s}(t)+X_{s} \cdot \dddot{s}(t)\right]-1 \leq 0 \tag{B.21}
\end{equation*}
$$

and its gradient with respect to the normalized segment durations is expressed as:

$$
\begin{equation*}
\frac{\partial C_{\ddot{X}}^{+}}{\partial \lambda^{T}}=\frac{1}{j_{\mathrm{x} \max }}\left[\left(X_{s s s s} \dot{s}^{3}+3 X_{s s s} \ddot{s} \ddot{s}+X_{s s} \dddot{s}\right) \frac{\partial s}{\partial \lambda^{T}}+3\left(X_{s s s} \dot{s}^{2}+X_{s s} \ddot{s}\right) \frac{\partial \dot{s}}{\partial \lambda^{T}}+3 X_{s s} \dot{s} \frac{\partial \ddot{s}}{\partial \lambda^{T}}+X_{s} \frac{\partial \ddot{s}}{\partial \lambda^{T}}\right] \tag{B.22}
\end{equation*}
$$

The gradients of the lower bound constraints have identical expressions to those of the upper bound constraints, with opposite sign (Eq.(B.14)). The gradients for the Y, Z, A and C axis constraints can be obtained by replacing the occurrences of ' X ' with the rest of axis variables in Eq. (B.18), (B.20), and (B.22).


[^0]:    Above constraints can be separated into individual inequalities;

