Time-Optimal Feedrate Planning for Freeform Toolpaths for Manufacturing Applications

by

Christina Qing Ge Chen

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2018

Author's Declaration

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

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

Abstract

Optimality and computational efficiency are two desired yet competing attributes of time-optimal feedrate planning. A well-designed algorithm can vastly increase machining productivity, by reducing tool positioning time subject to limits of the machine tool and process kinematics. In the optimization, it is crucial to not overload the machining operation, saturate the actuators' limits, or cause unwanted vibrations and contour errors. This presents a nonlinear optimization problem for achieving highest possible feedrates along a toolpath, while keeping the actuator level velocity, acceleration and jerk profiles limited. Methods proposed in literature either use highly elaborate nonlinear optimization solvers like Sequential Quadratic Programming (SQP), employ iterative heuristics which extends the computational time, or make conservative assumptions that reduces calculation time but lead to slower tool motion.

This thesis proposes a new feedrate optimization algorithm, which combines recasting of the original problem into a Linear Programming (LP) form, and the development of a new windowing scheme to handle very long toolpaths. All constraint equations are linearized by applying B-spline discretization on the kinematic profiles, and approximating the nonlinear jerk equation with a linearized upper bound (so-called 'pseudo-jerk'). The developed windowing algorithm first solves adjacent portions of the feed profile with zero boundary conditions at overlap points. Afterwards, using the Principle of Optimality, connection boundary conditions are identified that guarantee a feasible initial guess for blending the pre-solved adjacent feed profiles into one another, through a consecutive pass of LP.

Experiments conducted at the sponsoring company of this research, Pratt & Whitney Canada (P&WC), show that the proposed algorithm is able to reliably reduce cycle time by up to 56% and 38% in two different contouring operations, without sacrificing dynamic positioning accuracy. Benchmarks carried out with respect to two earlier proposed feedrate optimization algorithms, validate both the time optimality and also drastic (nearly 60 times) reduction in the computational load, achieved with the new method. Part quality, robustness and feed drive positioning accuracy have also been validated in 3-axis surface machining of a part with 1030 waypoints and 10,000 constraint checkpoints.

Acknowledgements

This research was sponsored by NSERC and Pratt & Whitney in Canada through grant number CRDPJ 462114 - 13. I would like to thank our technicians Mr. Robert Wagner, Mr. Neil Griffett and Mr. Andy Barber for their valuable assistance in my experiments.

I would like to thank Dr. Serafettin Engin, Dr. William Ferry, and Mr. Frederic Alexandre and at Pratt&Whitney Canada, Mr. Michael Ellis at Fanuc Canada in kindly providing valuable machine time, organizational and technical support needed for me to successfully validate my results.

Lastly, I would like to thank Dr. Mingyong Zhao from Tsinghua University for providing the mathematical expertise, and Professor Kaan Erkorkmaz for his continual guidance and support which is indispensable to the knowledge and skills I've gained throughout my Master's studies.

Dedication

To my husband Ryan

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Computer Numerical Control (CNC) is a machining technique in which a product can be manufactured automatically using a computer controlled machine tool. Once a program is designed, the CNC machine is able to run the process independently for an extended period of time, efficiently producing the desired parts with high accuracy, precision, and repeatability. CNC machines give the manufacturing industry the capability to mass produce complex 3D parts and surfaces continuously, free of human error. Hence, a well-designed feedrate optimization algorithm can immensely increase productivity, by reducing the cycle time, with a wide range of applications.

To prepare a part for mass production on a CNC machine, first, a solid or surface model is created of the desired product with Computer-aided Design (CAD), the model is then imported into a Computer-aided Manufacturing (CAM) software for toolpath planning. Within the CAM, an engineer is able to choose from a range of tools, stocks, and cutting operations (contouring, pocketing, engraving, etc.) that will determine the shape of the toolpath generated. Feedrate planning can be done within the CAM, or through a post processing software package that exclusively perform advanced feedrate optimization. The CNC and its controller are then able to take the axis level trajectories reconstructed from the toolpath geometry, tangential feedrates, and execute the point to point motions accordingly. An overview of the trajectory planning process is outlined in Figure 1-1.

This thesis extends a recent algorithm, which re-formulates the original problem with actuator velocity-, acceleration-, and jerk-limits, into a sub-optimal but convex form [1][2][3]. This form can then be reduced to a linear problem and efficiently solved by Linear Programing (LP). The new contribution extends the linearizing method, to allow for solving multiple sections of long toolpaths in parallel, such that the assembled solution performs close to the global one-shot method, while significantly reducing the computational time.

The organization of the thesis is as follows: literature review is presented in Chapter 2, followed by the proposed smooth toolplath generation algorithm in Chapter 3. Chapter 4 describes the LP formulation used to linearize the feedrate optimization problem, followed by the novel

windowing approach in Chapter 5. Chapter 6 presents the implementation and experimental results from the research. Finally, Chapter 7 concludes the thesis, and outlines the future work.

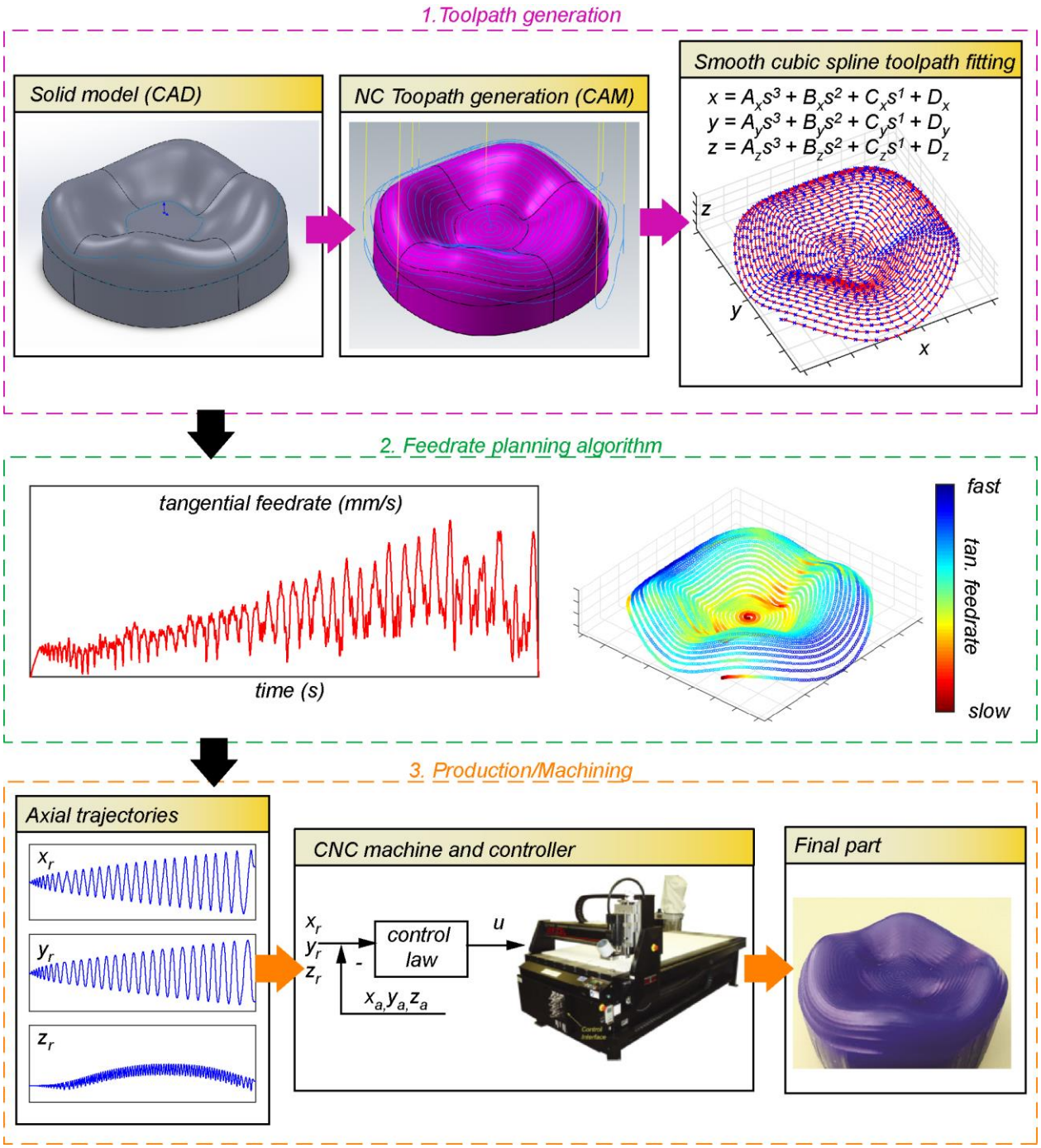

Figure 1-1 Overview the trajectory planning process for a CNC machine

Chapter 2
Literature Review

2.1 Introduction

The toolpaths generated from CAM software comprise lines and circular arcs positioned end to end to form a contour, continuous only in its geometric position, the connection points are known as 'corners'. For two geometrically similar toolpaths, where the first is smooth to the second geometric derivative (C^2 continuous), and the second is discretized into linear segments with numerous corners, if the tangential velocity profile is the same, the second toolpath will exhibit extreme fluctuations in its axial velocity, acceleration and jerk profiles. The spikes in the axial kinematic profiles will directly translate to excessive and erratic movements in the physical machine [4], potentially causing damage and reducing the operational life of the drives. To avoid this, feedrate optimization algorithms that consider the kinematics of the CNC will force the tangential feed to reach zero, whenever a corner is encountered, extending the cycle time for each corner. Thus, a smooth spline is usually fit to the original CAM toolpath, in order to prevent the tangential velocity from reaching zero. Currently, there exist a number of methods to smoothen a toolpath through spline or polynomial approximations [5][6][7][8], and corner rounding methods [9].

Typically, to execute the desired positions commanded by the CNC controller, each of the CNC's axial drives are independently powered by motors driving a screw shaft, with a table onto which stock material (blank) is secured. As the motor turns the screw, the table and blank will move along the screw shaft to achieve exact positioning. In this configuration, the maximum torque output of the motor directly determines the maximum velocity and acceleration that the CNC machine can reach. The maximum jerk limit is determined by the vibrations modes of the screw shaft system [10][11], usually determined by trial and error. In implementation, a reliable and robust feedrate planning algorithm should produce a velocity profile that avoid saturation of the motors, preventing damage to the machine hardware. Vibrations that are caused by the jerk component of the trajectory should also be limited, to retain part surface quality. At the same time, feed optimization needs to be able to handle long toolpaths for complex machining applications, such as point-milling of jet engine turbine blades [5].

2.2 Smooth Toolpath Fitting

In CNC machining, position coordinates along with feedrate information are given to the controller to be converted to movements in the axis drives. The tool tip is able to trace out a desired path from the combination of the axis movements. The first step for process optimization is to obtain the geometric toolpath information, in the forms given in Eq. (2.1), where r is the place holder for the three Cartesian coordinates x, y, z. s is a scalar parameter that spans the total arc displacement of the toolpath [6][12].

$$r(s), \quad r' = \frac{dr(s)}{ds}, \quad r'' = \frac{d^2 r(s)}{ds^2}, \quad r''' = \frac{d^3 r(s)}{ds^3} \qquad (2.1)$$
$$0 \le s \le L$$

The above derivatives are easily obtainable explicitly by discrete differentiation. Generally, toolpaths are given in the form of G-codes, comprised of linear and/or circular segments that are only continuous in positon (C^0 continuous), creating multiple corners. This becomes an issue for smooth feedrate interpolation since the axial machine kinematics (velocity v_r, acceleration a_r, and jerk j_r) are directly functions of the first, second and third geometric derivatives (r', r'', r''') with respect to the arc parameter s, as shown in Eq. (2.2) below, $\dot{s}, \ddot{s}$ and $\dddot{s}$ are the tangential velocity,

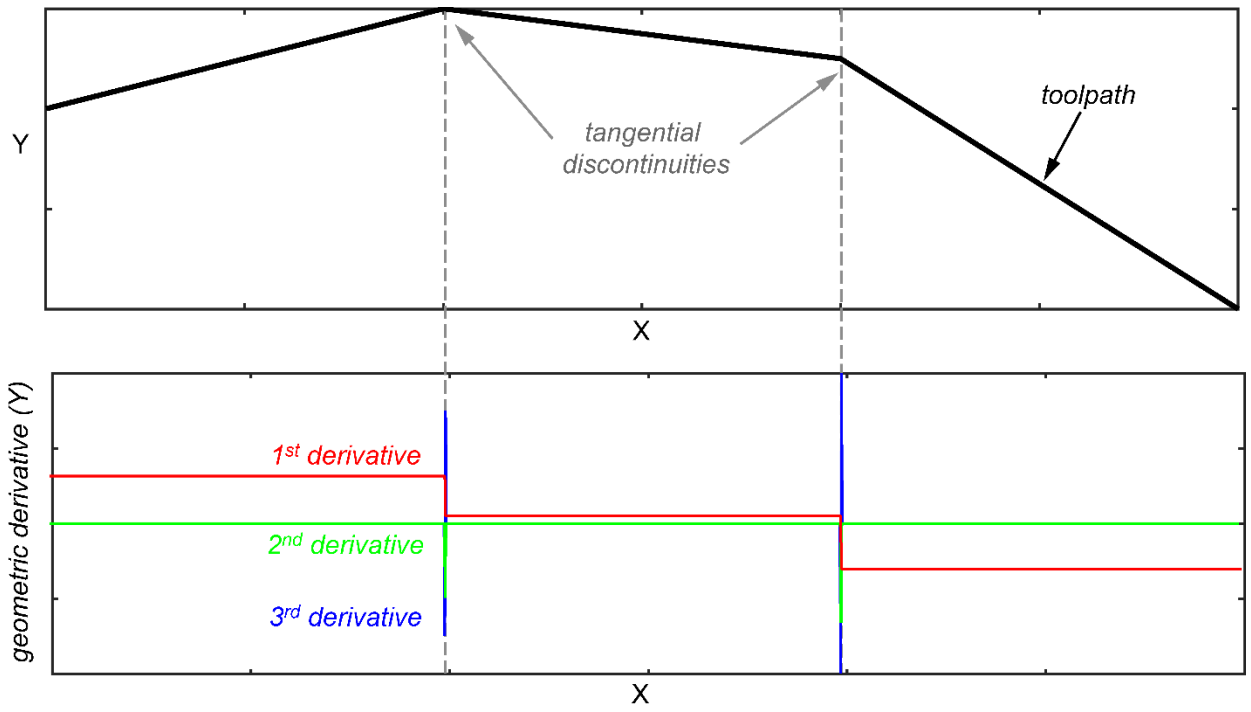

Figure 2-1 Toolpath with corners and its geometric derivative profiles

acceleration and jerk respectively. The full derivation of the equations can be found in Section 4.1 of the thesis.

$$\left.\begin{aligned} v_r &= r'\dot{s} \\ a_r &= r'\ddot{s} + r''\dot{s}^2 \\ j_r &= r'\dddot{s} + 3r''\dot{s}\ddot{s} + r'''\dot{s}^3 \end{aligned}\right\} \tag{2.2}$$

When the numerical derivatives are taken at the corners, their values quickly become unrestrained, as shown in Figure 2-1. Combined with Eq. (2.2), the feedrates will either become unbounded (if the optimization does not take into account the kinematics), or forced to come to a complete stop (i.e. $\dot{s}, \ddot{s}, \dddot{s} = 0$) to accommodate for the sudden change in toolpath curvature. Thus, the toolpaths as it is presented in the given state by G-codes are unsuitable for smooth feedrate interpolation. In literature, different types of splines [13][14][15][16][17] and polynomials [18][19] are usually fit with various degrees of continuity around the corner, regularly through pre-

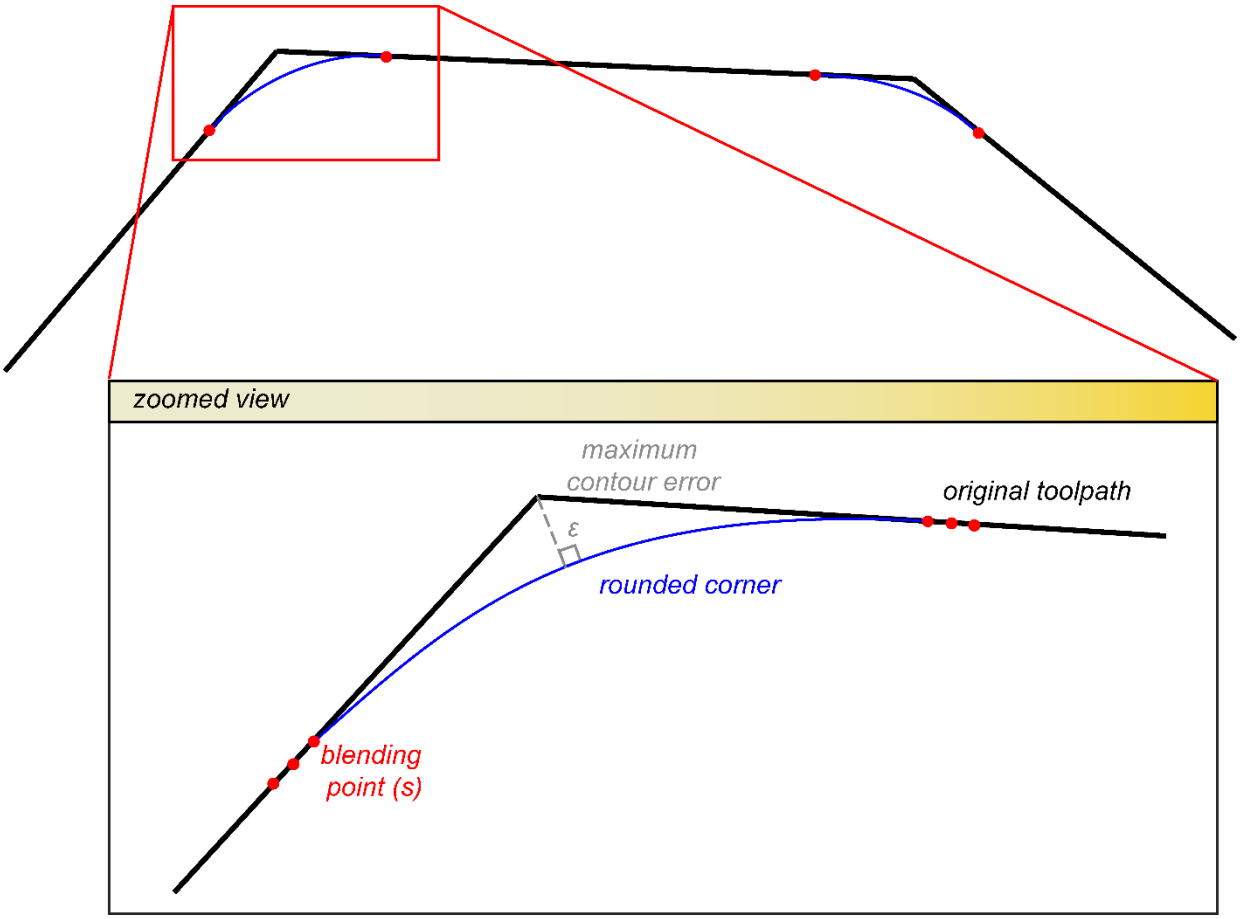

Figure 2-2 Long straight toolpath with local corner rounding

processing of the toolpath. More recently, splines are fit in real-time through advanced CNC controllers with spline interpolation capabilities [20]. Methods that do not require the pre-processing of the toolpath have also been proposed, such as correcting the motion of the drives directly when a corner is approached, by adjusting the acceleration and jerk profiles (proposed by Tajima et. al [21]), or using FIR filtering on the velocity command (proposed by Sencer et. al [22]). There are two main conditions where different smoothing methods can be applied. For toolpaths that mostly consists of long straight lines, common in pocketing operations [23], rounding only at the corners are sufficient. For toolpaths that are more 'curvy', i.e. consisting of many short linear segments, global smoothing is often used [24].

Figure 2-2 shows an example of a toolpath that consists of long linear segments with sharp corners. A curve is fit locally, symmetrical about the corner. In order to achieve smoothening, multiple blending points that lie on the two neighbouring linear segments are chosen according to the desired degree of continuity. For position, tangential, and curvature continuity, one, two and three blending point(s) that lie on the G-code toolpath are required, respectively. Maximum

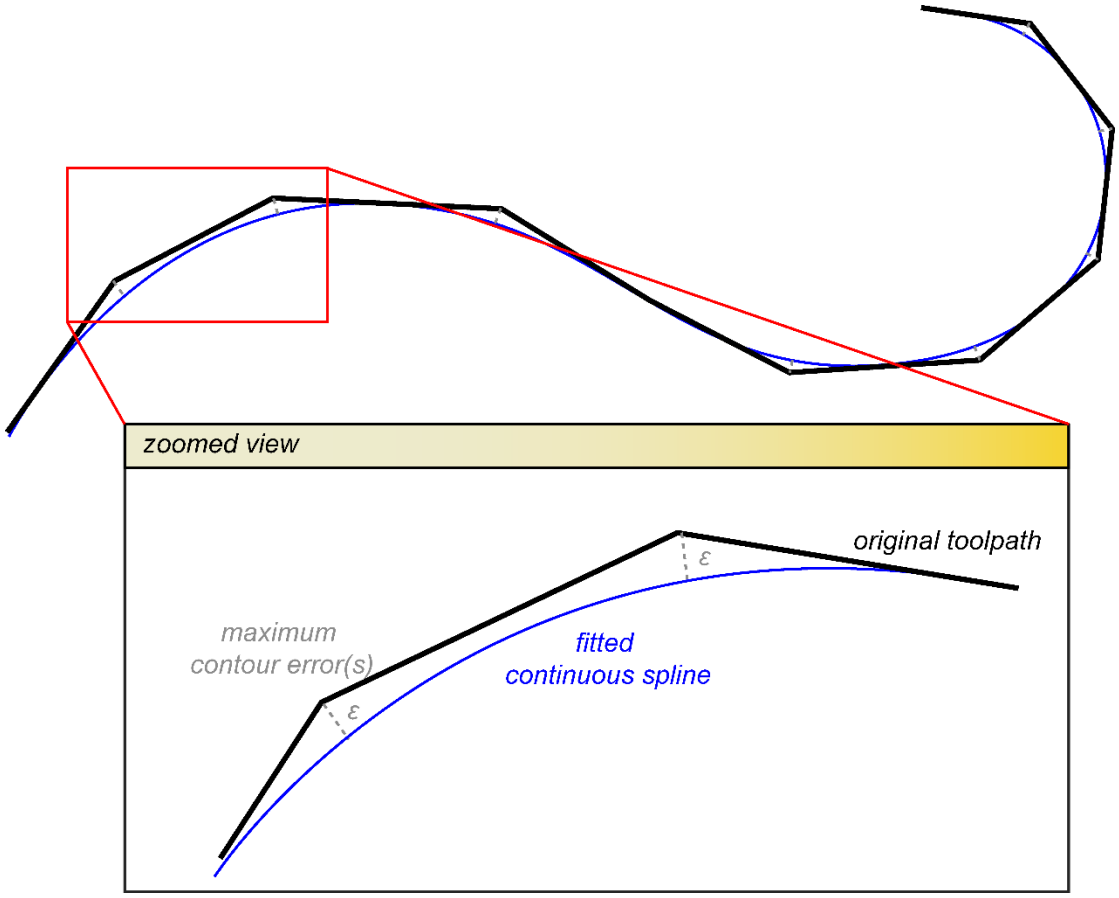

Figure 2-3 Short segmented toolpath fitted with global spline

contour error can also be controlled by the maximum closest perpendicular distance from the curve to the sharp corner on the original toolpath.

Figure 2-3 shows an example of a toolpath that is composed of numerous small linear segments, common in surface machining and engraving. Simple corner rounding methods are inappropriate in this case, as the corners often blend into each other due to the short length of the linear segments. Instead, fit points and/or control points can be chosen to fit a spline or polynomial through the original toolpath. As with the corner rounding methods, some contour errors are introduced, and can be limited by the shortest perpendicular distance according to part specifications.

2.3 Smooth and Time Optimal Feedrate Planning

Feedrate optimization began with basic methods which tries to predict the performance of the feed drives during the machining process, in attempt to avoid undesirable behaviour such as motor saturation, servo errors, and excessive vibrations. In medium to low-end CNC machines, constant feed trajectories are often used, where the CNC accelerates to a predetermined feed (usually found by trial and error, or through 'worst case' scenarios) at the start of the toolpath, continues with the same feed throughout, and then decelerates to a stop at the end. In the past, in an attempt to gain better control of the part quality through trajectory planning, researchers began to incorporate known physical factors from the machine and toolpath into the planning of machine processes.

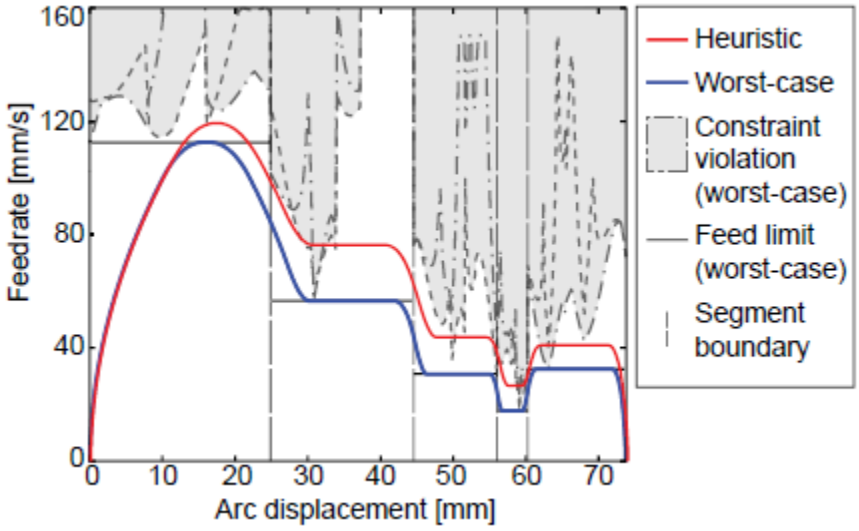

Figure 2-4 Comparison of feedrate profiles produced from Worst Case method and Heuristic method[33]

Chu et. al [25] identified specific part features such as convex, concave, ramp passes, etc., and determined the best feed for these features though experiments. Others have predicted and/or measured physical properties such as cutting forces, tool deflection, and chip thickness [26][27][28][29], and assigned feed values accordingly. Such methods can be non-optimal and time consuming due to the amount of machine test time required, for changing toolpaths for non-model based optimizations. Limitations on the trajectory kinematic and dynamic profiles such as torque and jerk have been proposed by Bobrow et. al [30] and Pritschow et. al [31], respectively, to increase part quality by avoiding motor saturations and excess vibration of the drives [6].

A computationally effective method to calculate feasible feedrate profiles to improve upon the constant feed method while considering the kinematic limits posed by the CNC machine, was proposed by Weck et al [32]. Here, a windowing method sets the feedrates of each segment based on the maximum feed limit in each window, the required computational load is minimal. However, the feedrates obtained from this method are conservative, due to always choosing the worst case scenario with regards to kinematic constraints. Erkorkmaz and Heng [33] improved upon this method by using a heuristic technique to iteratively search for maximum feeds that satisfies the kinematic constraints within each window, thereby reducing the cycle time at locations where the feed can be elevated. However, cycle time reduction is still limited by the size of the segments, since feedrates within a segment will plateau at a constant value, as can be seen in Figure 2-4. Further advancements can be made where the feedrate profiles stagnate. Other heuristic algorithms such as curve evolution (proposed by Sun et. al) [34] continuously modify the feedrate profile in small segments sequentially to achieve the lowest possible cycle time, through the use of B-splines, there is no stagnation of feed. Beudaert et al [35] proposed a Velocity Profile Optimization (VPOp) algorithm, in which the feedrate will always attempt to reach its maximum limit in the arc length parameter domain by traveling at maximum speed, until an unfeasible point is reached. Then, the algorithm back tracks to travel at a lower speed until all points are feasible. A Forward Projection method has also been proposed by Erkorkmaz [36] where back-tracking is not required, as the forward shooting method always ensures that the feedrates are able to come to a complete stop at all times. Iterative algorithm cannot guarantee the optimality of the solution, and some of the drawbacks include increased cycle time and/or computational time.

To achieve additional cycle time reduction, other techniques that seek to improve the feedrate profiles have been proposed. Sencer et al. [37] used a Sequential Quadratic Programming (SQP)

solver to modulate the control points of a B-spline feedrate profile, subject to the non-linear jerk constraint, resulting in a smooth profile with no stagnating feeds. Erkorkmaz and Altintas [38] presented a non-linear method to minimize feedrate fluctuations due to discrepancies in the spline parameter and arc length by describing the spline toolpath with respect to its spline length. Such algorithms improved the quality of the feedrate profiles at the expense of computational efficiency.

For handling long toolpaths, most methods proposed in literature use iterative techniques that rely on heuristic methods which can become time consuming. Usually, feedrates are constructed by first making a guess with an acceleration or jerk limited feed profile, then iterating based on constraint violations or the lack thereof. Continuity in the feedrate profile is achieved by sectioning the toolpath [39] and solving for each segment sequentially, looking ahead [36], backtracking after encountering a non-feasible point [35], or modifying local spline parameters [40]. Various authors have also employed some form of the two pass bi-directional scan approach [41][42][43], which optimizes the feedrate profile from both ends of the toolpath with free end conditions, then combining them together at the mid-intersection to from a feasible profile. For this thesis, the proposed algorithm is compared against the look ahead and backtracking method. Figure 2-5a) shows the forward projection look ahead algorithm [36]; the feedrate profile evolves as it moves along the path parameter from a jerk-limited dome trajectory, until an infeasible point is reached. The dome shaped feedrate profile is used to guarantee that the tool is always able to come to a complete stop at the end of the interpolation. Figure 2-6 shows the VPOp algorithm [35], at each step in the algorithm, the next maximum and minimum feed is calculated based on the current kinematics. The feed attempts to increase as fast as possible for each iteration point, until no solution can be found. Then, the algorithm back tracks to the point where choosing the minimum feed will allow the feedrate profile to reach to zero. The algorithm then modifies the subsequent points through the same iterations until a feasible profile is found.

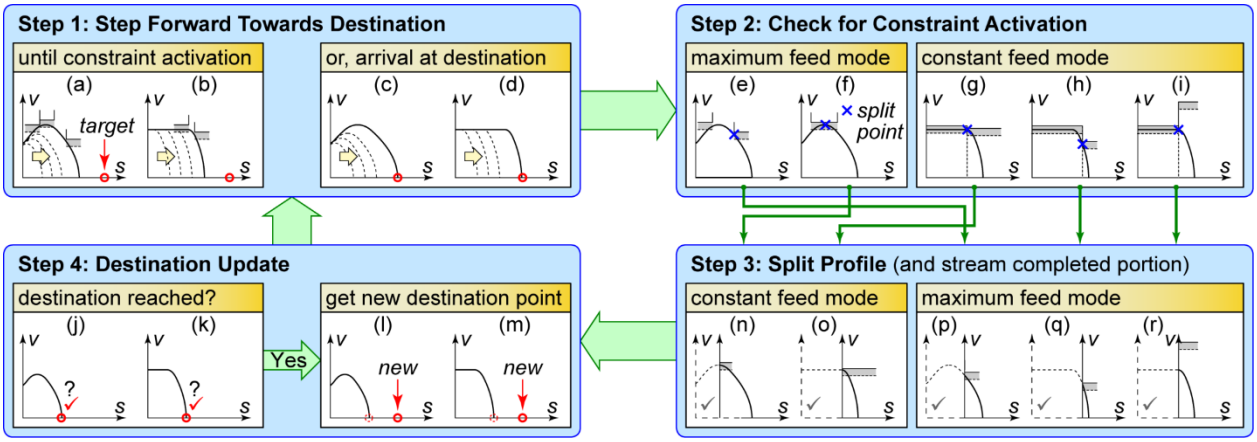

Figure 2-5 Forward projection feed planning algorithm [36]

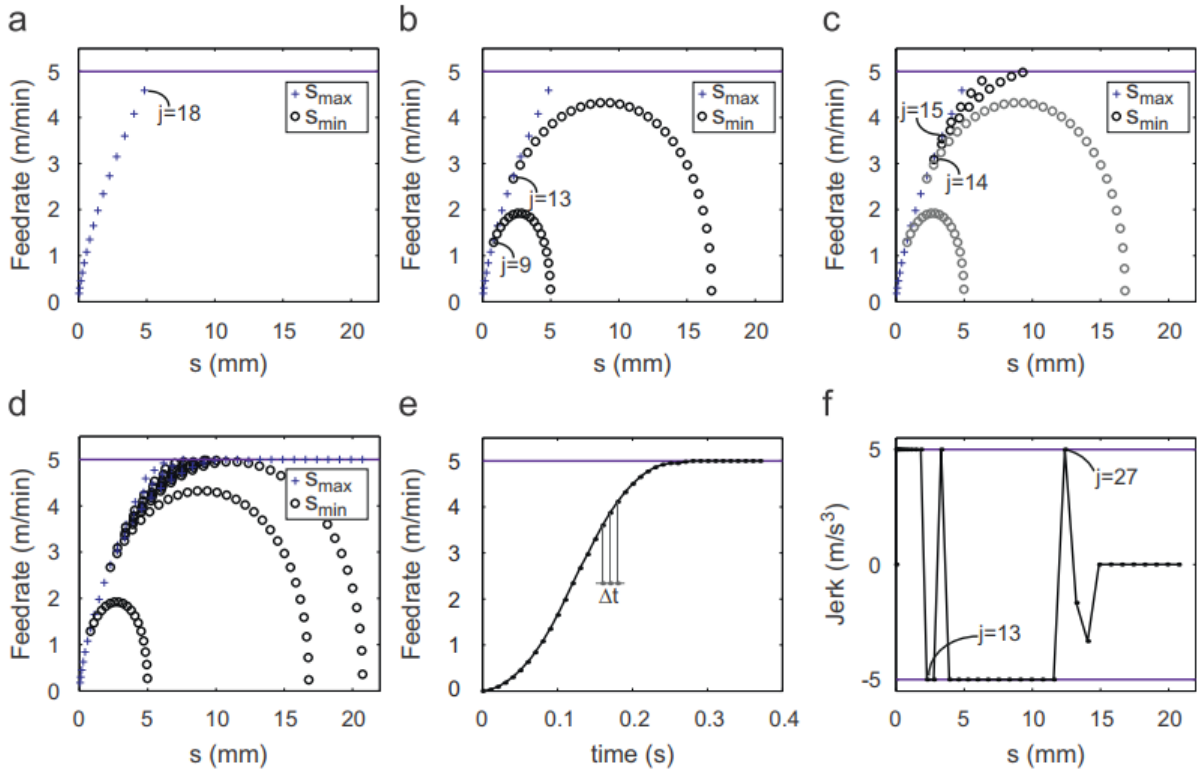

Figure 2-6 Velocity Profile Optimization (VPOp) algorithm [35]

2.4 Conclusions

In this chapter, an overview of past literature on the problems encountered during the feedrate planning process have been described, namely, smooth toolpath fitting and minimal time feedrate planning. In smooth toolpath fitting, various authors have proposed accurate and efficient methods for corner smoothing with limited contour deviations. In this thesis, a cubic spline global fit

formulation is chosen and will be presented in Chapter 3. In feedrate planning, currently there are two general approaches: heuristic algorithms or non-linear optimization solvers. The drawbacks are sub-optimality of the resulting feedrate profile, and heavy computational load, respectively. No standard method for extending a feedrate planning algorithm for a long toolpath currently exist. In this thesis, a new method for feedrate planning that reduces the non-linear kinematic constraint problems into a sub-optimal but linear form will be introduced in Chapter 4. A novel windowing approach that enables the processing of long toolpaths, without the need of backtracking, will be described in Chapter 5, with implementation and results in Chapter 6.

Chapter 3

Toolpath Generation from NC

3.1 Introduction

Generally, toolpath information is given in the format of G-Code, which is a common programming language understood and processed by CNC controllers. Figure 3-1 shows a portion of a sample G-Code performing a 2D contouring operation. The shape of the contour is created by joining segments of linear and circular arcs together sequentially, with each line in the G-code corresponding to a segment of the curve. When lines beginning with G01 are executed, the CNC will perform linear interpolation to the given coordinate from its current position. Lines beginning with G02 and G03 denote circular interpolation: clockwise (CW), and counter-clockwise (CCW) respectively, to the given coordinate, with arc centers given by I and J offsets from the current position. The resulting profile traced by commands from G-Codes are only guaranteed to be continuous in position at the segment transition points. For feedrate planning, the toolpath must be continuous to the second geometric derivative (C^2 continuous) in order for jerk to be bounded. A method of generating a cubic spline, which is continuous in both the first and second derivatives, within a set tolerance zone of the original toolpath is discussed in this chapter.

3.2 Cubic spline formulation

For a given toolpath, a set of *N+1* waypoints (points that lie on the toolpath) can be extracted

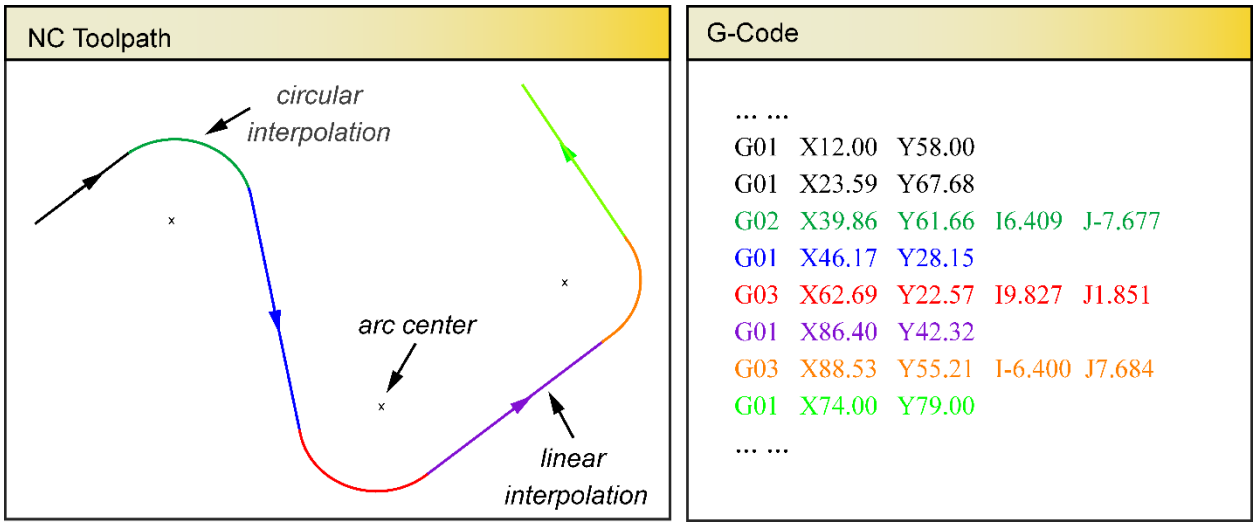

Figure 3-1 Sample toolpath and corresponding G-Code

such that *N* cubic splines can be fit in between each adjacent pair of points. For each spline segment, the position equation and its derivatives can be written as follows, and shown in Figure 3-2:

$$\begin{aligned} &r_i(u_i) = A_i u_i^{\,3} + B_i u_i^{\,2} + C_i u_i + D_i \\ &r_i'(u_i) = 3A_i u_i^{\,2} + 2B_i u_i + C_i \\ &r_i''(u_i) = 6A_i u_i + 2B_i \\ &0 \le u_i \le L_i \\ &i = 1,2,3,\ldots, N \end{aligned} \tag{3.1}$$

Above, *r* is a placeholder for the motion axes (*x* and *y* in for the toolpath example in Figure 3-2), *u* is the chord displacement along the toolpath, L_i is the chord length of the *i*th segment, and primes denote derivatives with respect to *u*. For each axis, there are *4N* unknowns (A_i,B_i,C_i,D_i). To make the resulting profile C^2 continuous, the end conditions (position, first and second derivatives) for each adjacent segments must be matched. Using Eq. (3.1), the following equations can be written for the *i*th segment.

Position end conditions (*2N* equations):

$$\begin{aligned} &r_i(0) = R_i \\ &r_i(L_i) = R_{i+1} \\ &i = 1,2,3\ldots N \end{aligned} \tag{3.2}$$

First derivative end conditions (*N-1* equations):

$$\begin{aligned} &r_i'(L_i) = r_{i+1}'(0) \\ &i = 1,2,3\ldots N-1 \end{aligned} \tag{3.3}$$

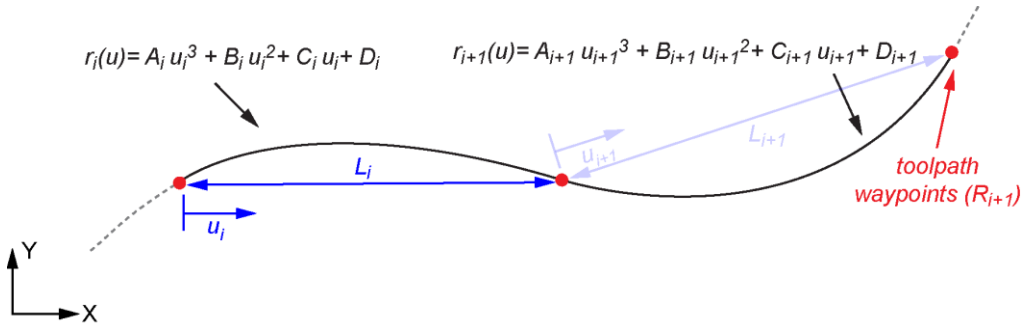

Figure 3-2 Cubic spline generation using chord displacements

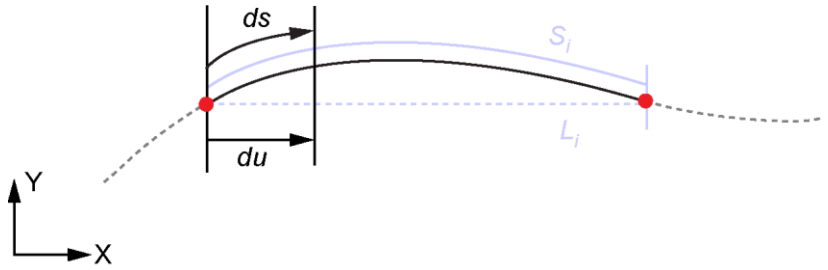

Figure 3-3 Difference between chord displacement (du) and arc length displacement (ds)

Second derivative end conditions *(N-1* equations):

$$r_i''(L_i) = r_{i+1}''(0) \qquad (3.4)$$
$$i = 1,2,3...N-1$$

The first and second derivatives at the beginning and end of the path can be assigned arbitrarily. Combining Eq. (3.2),(3.3),

(3.4), a total of *4N* equations can be used to solve the *4N* unknowns for a set of *N* unique cubic splines that passes through *N+1* waypoints. The generated toolpath is able to reliably and accurately provide path coordinates *x, y, z*, and their first, second and third derivatives with respect to *u* for all *u*.

3.3 Interpolation with minimal feedrate fluctuation

In feedrate optimization considering the kinematic constraints of the machines, velocity is specified along the tangent of the toolpath, therefore, derivatives with respect to the arc displacement *s* is required in the formulation of the kinematic constraints. The cubic spline formulated in Section 3.2 is parameterized with respect to the chord length *u*, since *du* and *ds* are not equivalent for toolpaths with non-zero curvature, as illustrated in Figure 3-3, a relationship between *u* and *s* needs to be found, so that the axis level displacements can be a written as a function of *s*. The following relationship is true for a 2D example:

$$\Delta s = \sqrt{\Delta x^2 + \Delta y^2} \qquad (3.5)$$

Δx and *Δy* can be written in terms of *u*:

$$\Delta x = \frac{dx}{du}\Delta u, \qquad \Delta y = \frac{dy}{du}\Delta u \tag{3.6}$$

As delta (Δ) approaches zero, and substituting Eq.(3.6) into Eq.(3.5):

$$ds = \sqrt{x'^2 + y'^2}\, du \tag{3.7}$$

x', y' can be analytically calculated as shown in Eq. (3.1), the relationship between s and u can therefore be found by integration of Eq. (3.7) discretely, and accessed through a look up table. For the remainder of the thesis, $x(u(s))$, $y(u(s))$, $z(u(s))$ will be referred to simply as $x(s)$, $y(s)$, $z(s)$.

3.4 Toolpath generation from iterative spline fitting

The curve resulting from the formulation in Section 3.2 may display fluctuation between the connection points in order to match the position boundary conditions, therefore, it may or may not follow the original toolpath to an acceptable degree of accuracy. The deviation from the nominal toolpath shape is dependent on the number of selected waypoints used in the spline fitting. If the distances between the points are long, the curve is free to vary excessively in-between. If the distances between the points are short, but positioned such that a smooth connection is unlikely,

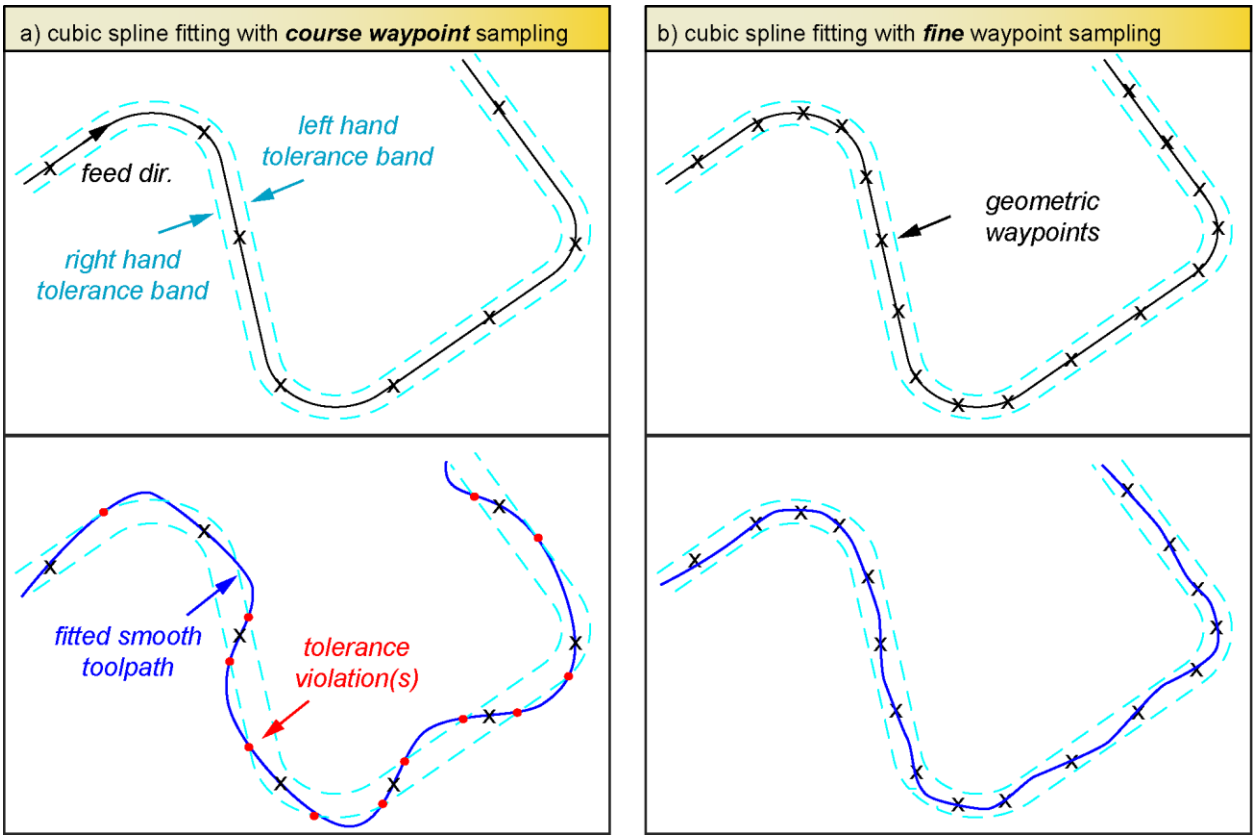

Figure 3-4 Spline fitting using course and fine waypoint sampling

the curve may oscillate in order to keep the end conditions satisfied. The latter can occur with noisy CAM data, where the way points are clustered and erratic. In aerospace applications considered in this thesis, the toolpaths are defined as connected line and arc segments, thus the first problem of waypoint distance being too large is more significant.

An iterative method is developed to find the number of way-points needed, such that the resulting cubic spline is within a certain degree of accuracy when compared to the original toolpath (Figure 3-4).

Step 1. Offset the original NC toolpath (Black) to create left and right hand tolerance bands (Cyan).

Step 2. Choose evenly spaced waypoints from the original toolpath defined in the G-Code, such that at least two waypoints are located on the shortest segment.

Step 3. Fit a cubic spline using waypoints selected from Step 2, with the method described in Section 3.2 (Blue).

Step 4. Check for intersections between the cubic spline and tolerance band (Red). If intersections exist, then at least a portion of the cubic spline will be outside of the accepted tolerance zone.

Step 5. If intersections exist, increase the number of waypoints by some amount, usually 10~20%.

Step 6. Iterate Steps 3-5 until there are no more intersections between the cubic spline and tolerance band.

The resulting cubic spline will almost always be guaranteed to be bounded by the right and left hand tolerances specified by the user. In some cases, with a particularly low tolerance, corners can be too sharp to be fit without violating the bounds, as shown in Figure 3-5a. In the cubic spline fitting algorithm, increasing the number of waypoints will not effectively reduce the amount of tolerance violations. In such cases, the toolpath can be split, such as in Figure 3-5b. The feedrate optimization algorithm will force the CNC to come to a complete stop at the corner due to the discontinuous geometric derivatives. The toolpath is essentially split into two independent toolpaths at the corner, and the cycle time is increased locally for improved geometric accuracy.

3.5 Conclusions

In this chapter, a method of fitting a cubic spline to a G-code toolpath is described. The algorithm is able to take any 2D toolpath and produce a curve that is within a certain degree of accuracy to the original contour. The new curve and its geometric derivatives are suitable for use in the feedrate planning algorithm described in the next chapter, that produces jerk limited trajectories.

.

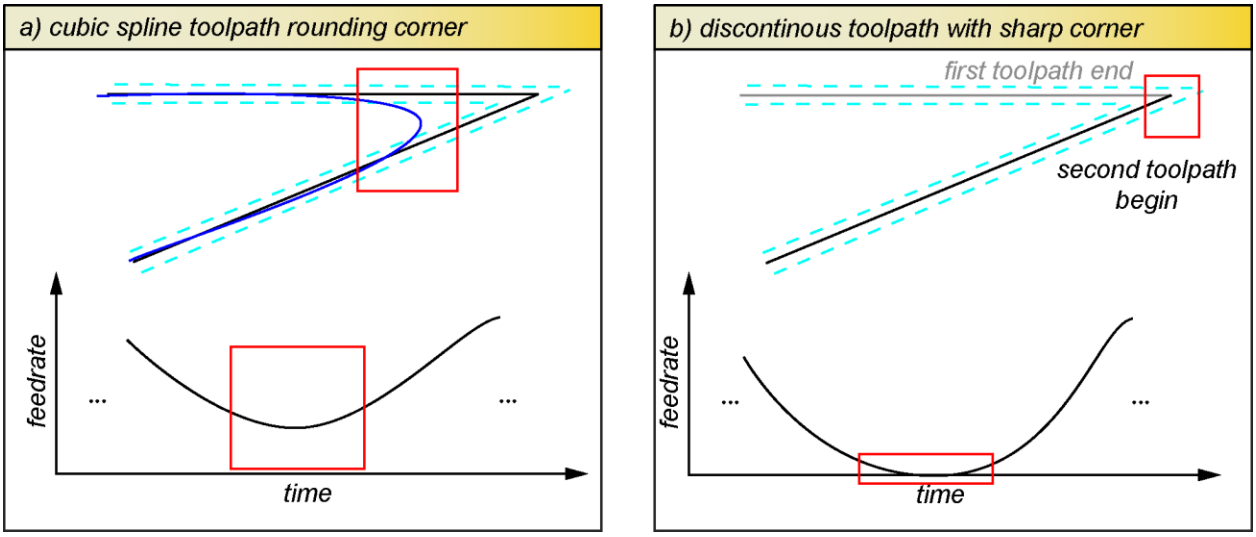

Figure 3-5 Splitting a toolpath when a toolpath corner is too sharp

Chapter 4
Feedrate Optimization using Linear Programming

4.1 Introduction and Problem Formulation

Given a set of waypoints on a toolpath, a toolpath is first discretized, such that the third order parametric derivative can be evaluated for all *s* as described in Chapter 3:

$$\left.\begin{aligned} &r(s) = (x(s), y(s), z(s)) \\ &0 \le s \le L \end{aligned}\right\} \tag{4.1}$$

Above, *s* is the toolpath displacement parameter, corresponding to the spline arc length, and *L* is the total length of the toolpath. Then, the axis level velocity, acceleration and jerk that the CNC machine experiences while traversing the toolpath, can be found by taking the time domain derivative of the position functions. Using the chain rule, the simplified form of the expressions are shown in Eq. (4.2). Primes denote a derivative with respect to the arc parameter *s*, overhead dots denote derivative with respect to time *t*.

$$\left.\begin{aligned} v &= \frac{dr}{dt} = \frac{dr}{ds}\frac{ds}{dt} = r'\dot{s} \\ a &= \frac{d^2 r}{dt^2} = \frac{d}{dt}(r'\dot{s}) = \frac{d(\dot{s})}{dt}r' + \frac{d(r')}{dt}\dot{s} = \frac{d(\dot{s})}{dt}r' + \frac{d(r')}{ds}\frac{ds}{dt}\dot{s} = r'\ddot{s} + r''\dot{s}^2 \\ j &= \frac{d^3 r}{dt^3} = \frac{d}{dt}(r'\ddot{s} + r''\dot{s}^2) = \frac{d\ddot{s}}{dt}r' + \frac{dr'}{dt}\ddot{s} + \frac{d(\dot{s}^2)}{dt}r'' + \frac{dr''}{dt}\dot{s}^2 \\ &= \dddot{s}r' + \frac{dr'}{ds}\frac{ds}{dt}\ddot{s} + 2\dot{s}\frac{d(\dot{s})}{dt}r'' + \frac{dr''}{ds}\frac{ds}{dt}\dot{s}^2 = r'\dddot{s} + 3r''\dot{s}\ddot{s} + r'''\dot{s}^3 \end{aligned}\right\} \tag{4.2}$$

$\dot{s}, \ddot{s}$ and $\dddot{s}$ are the parametric velocity, acceleration, and jerk respectively. r', r'' and r''' are the axis level geometric derivatives of the toolpath from Eq. (3.1).

The ultimate goal for feedrate planning is to reduce the total cycle time. Intuitively, the average velocity should be maximized. Given Eq. (4.2), an optimization problem that seeks to maximize average feedrate $\dot{s}$ along the toolpath can be formed using the following constraint inequalities,

where $v_{\max}$, $a_{\max}$ and $j_{\max}$ are the axis level kinematic limits of the CNC, usually dependent on the motor saturation limits and vibration as well as position tracking bandwidth of the machine.

$$\left.\begin{aligned} &\text{maximize} \int_0^L \dot{s}\, ds \\ &subject\,to: \\ &|\, v_{\max} \,| \geq |\, r'\dot{s} \,| \\ &|\, a_{\max} \,| \geq |\, r'\ddot{s} + r''\dot{s}^2 \,| \\ &|\, j_{\max} \,| \geq |\, r'\dddot{s} + 3r''\dot{s}\ddot{s} + r'''\dot{s}^3 \,| \\ &0 \leq s \leq L \end{aligned}\right\} \tag{4.3}$$

The problem as it is currently expressed is challenging to solve. Since *t* is unknown, any derivatives with respect to time: $\dot{s}, \ddot{s}$ and $\dddot{s}$ cannot be computed explicitly. Thus it is more convenient to reformulate the equations to be strictly in term of *s*. Therefore, a new substitution parameter q, is used, defined as the square of the parametric velocity $\dot{s}^2$. Through the substitution of *q* and its parametric derivatives into Eq. (4.3), elimination of all terms with respect to time can be achieved. First, derivatives of *q* with respect to *s* are obtained using the chain rule:

$$\left.\begin{aligned} &q = \dot{s}^2 \\ &q' = \frac{d(\dot{s}^2)}{ds} = 2\dot{s}\frac{d\dot{s}}{ds} = 2\dot{s}\frac{d\dot{s}}{dt}\frac{dt}{ds} = \frac{2\dot{s}\ddot{s}}{\dot{s}} = 2\ddot{s} \\ &q'' = 2\frac{d\ddot{s}}{ds} = 2\frac{d\ddot{s}}{dt}\frac{dt}{ds} = \frac{2\dddot{s}}{\dot{s}} \end{aligned}\right\} \tag{4.4}$$

Through variable manipulation, q, q' and q'' can be expressed as functions of $\dot{s}, \ddot{s}$ and $\dddot{s}$. Inversely, all terms that are derivatives with respect to time in Eq. (4.3) can be re-written using the expressions in Eq. (4.4):

$$\left.\begin{aligned} &\dot{s} = \sqrt{q}, \qquad \dot{s}^2 = q, \qquad \dot{s}^3 = \dot{s}\dot{s}^2 = q\sqrt{q} \\ &\ddot{s} = \frac{1}{2}q', \qquad \dddot{s} = \frac{1}{2}q''\dot{s} = \frac{1}{2}q''\sqrt{q}, \qquad \dot{s}\ddot{s} = \frac{1}{2}q'\dot{s} = \frac{1}{2}q'\sqrt{q} \end{aligned}\right\} \tag{4.5}$$

Re-writing Eq. (4.3) using the substitutions in Eq. (4.5), and considering that q is always positive by its definition, the equivalent optimization problem without any time derivative term

becomes:

$$\left.\begin{aligned} &\text{maximize} \int_0^L q\, ds \\ &subject\,to \\ &|v_{\max}| \geq |r'|\sqrt{q} \\ &|a_{\max}| \geq |r''q + \frac{1}{2}r'q'| \\ &|j_{\max}| \geq |(r'''q + \frac{3}{2}r''q' + \frac{1}{2}r'q'')|\sqrt{q} \\ &0 \leq s \leq L \end{aligned}\right\} \tag{4.6}$$

The presence of terms q', q'' and $\sqrt{q}$ in the constraint equations requires the use of non-linear solvers for computing the solution. This is undesirable since such solvers generally take a long time to reach convergence with no guarantee of optimality. Section 4.2 and Section 4.3 will discuss the methods adapted in this thesis for linearizing the problem to enable the use of fast and robust Linear Programming (LP) solvers, proposed by Fan et. al [2] .

4.2 B-spline Discretization and Linearizing the Velocity and Acceleration Equations

Typically, in optimal trajectory planning, the constraints are considered only at certain discrete points along the toolpath. This allows for matrix-style representation of the inequalities in Eq. (4.6). This is achieved by evaluating the constraint expressions at selected locations along *s*, as shown in Figure 4-1. Here, *K* represents the total number of discretization points.

Given Eq. (4.6), in the velocity constraint expression, the non-linearity can be removed by squaring both sides. $v_{\max}$ and r' are constants independent of feedrate. In the acceleration and jerk constraint equations, q' and q'' terms can be linearized by using discrete differentiation methods

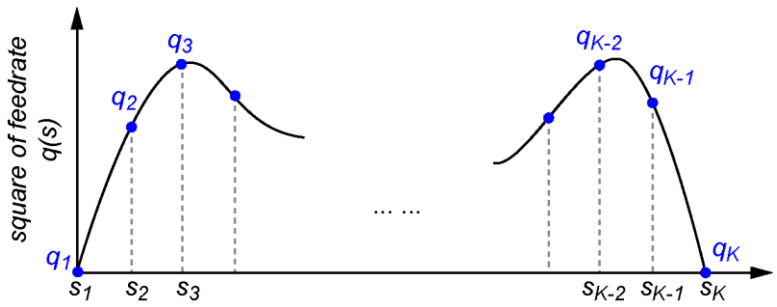

Figure 4-1 Discretization of the feedrate squared (q) profile

with respect to *s*, or by constructing the profile for *q* in the form of a second order B-Spline. Both methods were tried in this thesis research, and it was found that the B-spline formulation provided superior numerical properties over the discrete differentiation methods, in terms of smoothness of the spline and stability of the optimization implementation.

De Boor's algorithm [44][45] is used to generate the *q* profile as a B-spline, where the value of the function is a linear combination of neighboring control point values and their basis functions. Figure 4-2 shows an example of how q_k can be calculated for any s_k using this method. $N_i(s)$ are the *K* basis functions of a second order B-spline. At any point along *s*, the sum of the basis function values is equal to one. a_i are the control points, weighted by the value of the basis function depending on the spline's current position along *s*. a_i are used in optimization to adjust the shape of the resulting spline. The knot vector $S_n = [0,0,0,S_1,S_2,\ldots,S_{K-3},L,L,L]$ spans the arc length of the toolpath, and defines the beginning and end of each basis function $N_i(s)$. For a second order B-

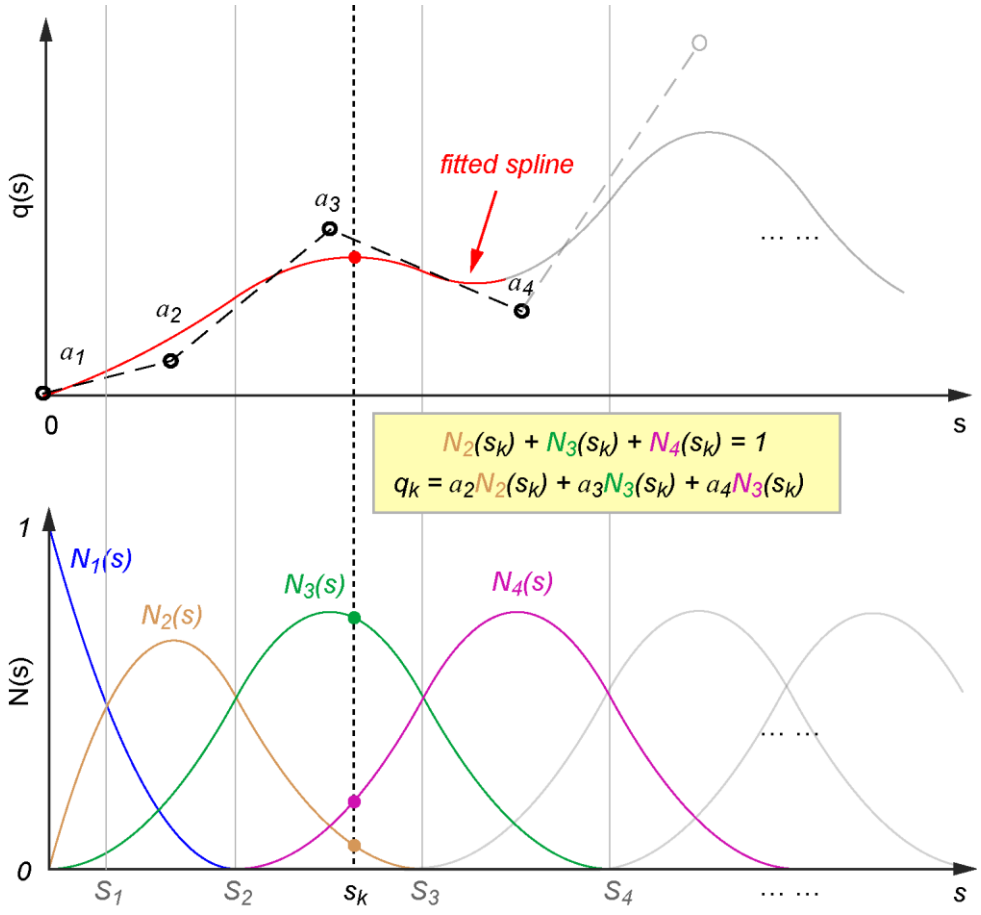

Figure 4-2 Second order B-spline formulation using basis functions

spline, the value of the spline at any given point (such as s_k) is affected by the adjacent three control points and their basis function. Therefore, any knot value that is repeated three times in the knot vector will cause the spline to pass through the control points at these locations exactly. For feedrate interpolation, it is important to control the exact value at the start and end of the profile, since they are usually explicitly specified by practical requirements of the process (Example: rest conditions).

In short form, the q profile and its derivatives can be written as shown in Eq.(4.7). Note that now, q' and q'' are also linear with respect to the control points a_i:

$$q(s)=\sum_{i=1}^{K} N_{i,2}(s)\cdot a_i, \qquad q'(s)=\sum_{i=1}^{K} N'_{i,2}(s)\cdot a_i, \qquad q''(s)=\sum_{i=1}^{K} N''_{i,2}(s)\cdot a_i \tag{4.7}$$

The values of the basis functions and their derivatives for a 2nd order B-spline are calculated using only the knot vector S_n, and the current position s_k for all s, as shown in Eq. (4.8). These calculations need only to be performed once, while fitting the toolpath geometry, and the results can be stored in a constant array. The p subscript indicates the order of the B-spline.

$$\left.\begin{aligned}
N_{i,0}(s) &= \begin{cases} 1 & if\ S_n(i)\le s\le S_n(i+1) \\ 0 & otherwise \end{cases}, \quad i=0,\ldots,K+3 \\
N_{i,p}(s) &= \frac{u-S_n(i)}{S_n(i+p)-S_n(i)}N_{i,p-1}(s)+\frac{S_n(i+p+1)-s}{S_n(i+p+2)-S_n(i+1)}N_{i+1,p-1}(s), \quad i=0,\ldots,K+1 \\
N'_{i,p}(s) &= \frac{p}{S_n(i+p)-S_n(i)}N_{i,p-1}(s)-\frac{p}{S_n(i+p+1)-S_n(i+1)}N_{i+1,p-1}(s), \quad p=1,2 \\
N''_{i,p}(s) &= \frac{p}{S_n(i+p)-S_n(i)}N'_{i,p-1}(s)-\frac{p}{S_n(i+p+1)-S_n(i+1)}N'_{i+1,p-1}(s),
\end{aligned}\right\} \tag{4.8}$$

Then, q, q' and q'' profiles can then be written in matrix form:

$$\left.\begin{aligned}
\begin{bmatrix} q_1 \\ \vdots \\ q_K \end{bmatrix} &= \begin{bmatrix} N_{1,2}(s_1) & \cdots & N_{K,2}(s_1) \\ \vdots & \ddots & \vdots \\ N_{1,2}(s_K) & \cdots & N_{K,2}(s_K) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_K \end{bmatrix}, \quad \begin{bmatrix} q'_1 \\ \vdots \\ q'_K \end{bmatrix} = \begin{bmatrix} N'_{1,2}(s_1) & \cdots & N'_{K,2}(s_1) \\ \vdots & \ddots & \vdots \\ N'_{1,2}(s_K) & \cdots & N'_{K,2}(s_K) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_K \end{bmatrix} \\
\begin{bmatrix} q''_1 \\ \vdots \\ q''_K \end{bmatrix} &= \begin{bmatrix} N''_{1,2}(s_1) & \cdots & N''_{K,2}(s_1) \\ \vdots & \ddots & \vdots \\ N''_{1,2}(s_K) & \cdots & N''_{K,2}(s_K) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_K \end{bmatrix}
\end{aligned}\right\} \tag{4.9}$$

Replacing instances of q and its derivatives in Eq. (4.6) with the matrix representation in Eq. (4.9) makes the acceleration inequality $|a_{\max}| \geq |r''q + \frac{1}{2}r'q'|$ linear with respect to control points a_i, which can then be used as the free variable in the optimization to maximize the values of the q curve.

Along with zero boundary conditions (v, $a = 0$), the algorithm proposed in the thesis (Section 5.2) also requires other boundary conditions to be respected, in order to connect two feedrate profiles together at a specific location along s. To keep the feedrate profile smooth, the axis level velocity and acceleration must be equal at the location where two feedrate profiles meet. From Eq. (4.2), it can be seen that $\dot{s}$ and $\ddot{s}$ dictate the value of the axis level velocity and acceleration for all s, $\dot{s}$ and $\ddot{s}$ are in turn determined only by q and q' from Eq. (4.5). To fix the value of q and q' at any ith control point, the following equality constraints are added to the optimization problem:

$$\left.\begin{aligned}
q_i &= \begin{bmatrix} N_{1,2}(s_i) & \cdots & N_{K,2}(s_i) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_K \end{bmatrix}, \\
q'_i &= \begin{bmatrix} N'_{1,2}(s_i) & \cdots & N'_{K,2}(s_i) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_K \end{bmatrix}
\end{aligned}\right\} \tag{4.10}$$

4.3 Linearization using Pseudo-Jerk

From Eq. (4.6), recall that the jerk inequality has the following form:

$$|j_{\max}| \geq |(r'''q + \frac{3}{2}r''q' + \frac{1}{2}r'q'')|\sqrt{q} \tag{4.11}$$

From the previous section, terms inside the brackets can be linearized with respect to the control points a_i by replacing q, q' and q'' with their equivalent linear matrix representation. However, no such linearization exists for $\sqrt{q}$. To linearize the jerk inequality, an adequate substitution for $\sqrt{q}$ must be found such that the jerk constraint will still hold for all s.

If an approximate upper bound for the q profile can be calculated, denoted as q^*, it can be used as a constant for replacing the q term within the square root. This would linearize the jerk equation. The new constraint with q^* can be defined as pseudo-jerk $\tilde{j}$:

$$\tilde{j} = (r'''q + \frac{3}{2}r''q' + \frac{1}{2}r'q'')\sqrt{q^*} \tag{4.12}$$

Since q is always positive by its definition in Eq. (4.4), it can be concluded that if $q^* \geq q$, then $|\tilde{j}| \geq |j|$ for all s, as seen in Figure 4-3. Therefore, if pseudo jerk $\tilde{j}$ is kept under the maximum jerk limit $j_{\max}$, then the real jerk j is bounded by $j_{\max}$ as well. Solving the following optimization problem typically yield a suitable q^* profile that functions as a tight upper bound approximation for q.

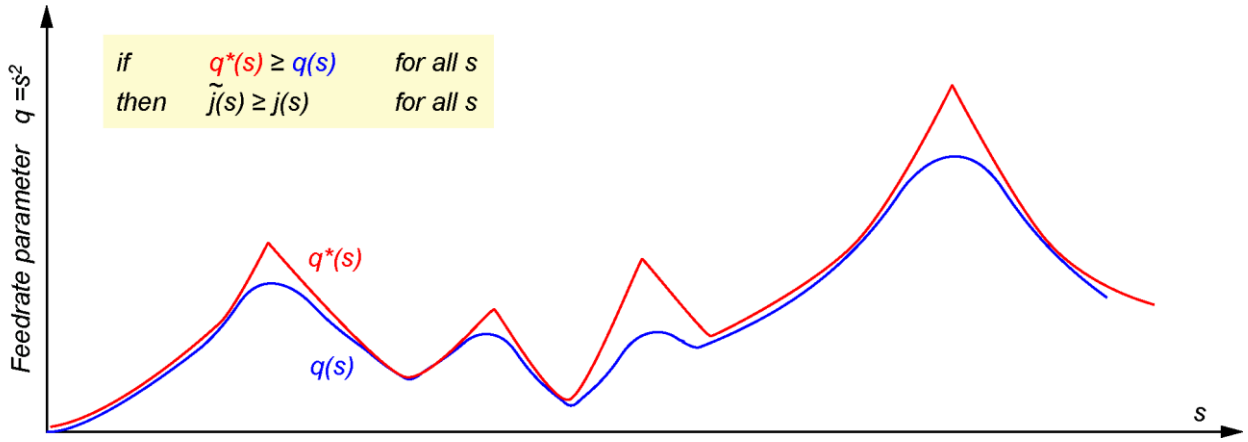

Figure 4-3 *q as an upper bound for the expected optimal solution *q***

$$\left.\begin{array}{l} \text{maximize } \sum q_i^* \\ subject\ to \\ |v_{i,\max}| \geq |r_i'| \sqrt{q_i^*} \\ |a_{i,\max}| \geq |r_i'' q_i^* + \frac{1}{2} r_i' q_i'^*| \\ 0 \leq s \leq L \\ i = 1, \ldots, K \\ \text{optional constraints:} \\ q_{process} \geq q^* \\ q_{BC} = q^* \end{array}\right\} \quad (4.13)$$

Note that Eq. (4.13) is the same problem as the one presented in Eq. (4.6), with the jerk inequality removed. Intuitively, removing constraints from an optimization problem will cause the new solution to have higher values than the original solution, since the profile is able to increase freely at locations where the removed constraint(s) were active. In [46], it has been shown that in an optimal feedrate generation problem whose constraints involve the velocity and acceleration only, the optimal solution is unique and is maximum among all possible feedrate solutions at any time. Therefore, the feedrate profile found in the velocity-acceleration limited problem can be used as an upper bound on q.

In the optimization problem, $q_{process} \geq q^*$ is the constraint on the feedrate profile that arises from other physical limitations of the CNC machine, such as cutting load, chip load, or spindle speed. These limits may be obtained from other machining optimization software. $q_{BC} = q^*$ is the boundary conditions imposed on the feedrate profile, from process or algorithm requirements. After solving q^* from the problem in Eq. (4.13), the full linearized problem, including all constraints, becomes:

$$\left.\begin{array}{l}
\text{maximize } \sum q_i \\
\textit{subject to} \\
|\, v_{i,\max} \,| \geq |\, r_i' \,| \sqrt{q_i} \\
|\, a_{i,\max} \,| \geq |\, r_i'' q_i + \frac{1}{2} r_i' q_i' \,| \\
|\, j_{i,\max} \,| \geq |\, (r_i''' q_i + \frac{3}{2} r_i'' q_i' + \frac{1}{2} r_i' q_i'') \,| \sqrt{q_i^*} \\
0 \leq s \leq L \\
i = 1, \ldots, K \\
\text{optional constraints :} \\
q_{process} \geq q \\
q_{BC} = q
\end{array}\right\} \qquad (4.14)$$

Eq. (4.13) and Eq. (4.14) are solved sequentially to provide the final profile for q. In cases where the jerk is not the primary constraint on the shape of the q profile, the resulting q profile will be very close to the optimal solution found of the original problem by a non-linear solver. However, in cases where the jerk limit tends to be the most dominant, linearization via the use of pseudo-jerk does bring forth a certain amount of sub-optimality. This sub-optimality needs to be further investigated and quantified in future research.

4.4 Extension of Pseudo-Jerk formulation

For toolpaths where the jerk is the prominent factor in shaping the feedrate, the q^* profile calculated from the first optimization step will be substantially higher than the q profile computed in the second step. This will cause the pseudo jerk to become an over-estimation of the real jerk, reducing the optimality of the solution such that the actual jerk never reaches the CNC jerk limit. Thus, as an approximate constraint, the first term of the jerk equation, $|\, j_{i,\max} \,| \geq |\, r_i''' q_i^* \,| \sqrt{q_i^*}$ is added to the first optimization step. The first term, while linear, is able to capture the majority of the effect of feed rate profile $\sqrt{q}$ on the axis level jerk. To keep $|\, \tilde{j} \,| \geq |\, j \,|$ true, an extra constraint $q^* \geq q$ is added to the second optimization step. The extended pseudo jerk formulation is shown in Eq. (4.15).

First optimization: **Second Optimization:**

$$\left.\begin{array}{l} \text{maximize } \sum q_i^* \\ \textit{subject to} \\ |v_{i,\max}| \geq |r_i'| \sqrt{q_i^*} \\ |a_{i,\max}| \geq |r_i'' q_i^* + \frac{1}{2} r_i' q_i'^*| \\ |j_{i,\max}| \geq |r_i''' q_i^*| \sqrt{q_i^*} \\ 0 \leq s \leq L \\ i = 1,\ldots,K \\ \text{optional constraints:} \\ q_{process} \geq q^* \\ q_{BC} = q^* \end{array}\right\} \quad \left\{\begin{array}{l} \text{maximize } \sum q_i \\ \textit{subject to} \\ |v_{i,\max}| \geq |r_i'| \sqrt{q_i} \\ |a_{i,\max}| \geq |r_i'' q_i + \frac{1}{2} r_i' q_i'| \\ |j_{i,\max}| \geq |(r_i''' q_i + \frac{3}{2} r_i'' q_i' + \frac{1}{2} r_i' q_i'')| \sqrt{q_i^*} \\ q^* \geq q \\ 0 \leq s \leq L \\ i = 1,\ldots,K \\ \text{optional constraints:} \\ q_{process} \geq q \\ q_{BC} = q \end{array}\right. \tag{4.15}$$

The two optimization problems, when written in matrix form, will enable the use of fast and robust discrete LP solvers, where the tangential feedrate parameter *q* is solved for in terms of *s*.

4.5 Conclusions

In this chapter, through the formulation of the feedrate square profile as a B-spline, the introduction of the new feedrate parameter *q*, and the use of the estimated pseudo-jerk $\tilde{j}$, a method for linearizing the non-linear optimization problem for minimal time feedrate planning, considering the CNC machine kinematics, has been presented. Although sub-optimal, the new formulation gives the advantage of being able to employ fast and robust LP solvers in reaching a solution for a feasible feedrate profile. Chapter 5 will present the novel windowing algorithm developed as a part of this thesis on the application of the LP formulation for indefinitely long toolpaths. Chapter 6 will present both simulation and experimental results and document the performance of the proposed algorithms against past literature.

Chapter 5
Windowing Algorithm for Long Toolpaths and Other Improvements and Implementation Measures

5.1 Introduction

For optimization algorithms, the computational time grows significantly with the increasing number of constraints evaluated at each optimization step. For the two stage LP formulation described in Chapter 4, for a single axis, a total of 9 inequalities (2 for velocity, 3 for jerk and 4 for acceleration) are evaluated for each q_i. Therefore, for K points on the feedrate profile, a total of at least 9K equations will be evaluated. In matrix representation, the computational time grows as a polynomial order of 4.4 with matrix size, this can be seen by performing a runtime test on a sample toolpath with increasing optimization control points K. From the curve fit in Figure 5-1, it is obvious that the computational time can quickly escalate beyond the capability of the processor as K grows large. It is important to note that for a given toolpath, K may not always be directly related to its length. For a short toolpath composed of numerous curves with small radii (i.e. sharp corners), the number of control points may be greater than that of a straight but lengthier curve, since more control points are required to modulate a feedrate profile with faster rates of change. For real-time applications, it is important that the algorithm is able to work quickly for any given toolpath, with the capability to execute with an indefinite amount of control points.

This Chapter describes a novel approach to reduce the computational time in processing long toolpaths using the LP feed optimization solution presented in Chapter 4.

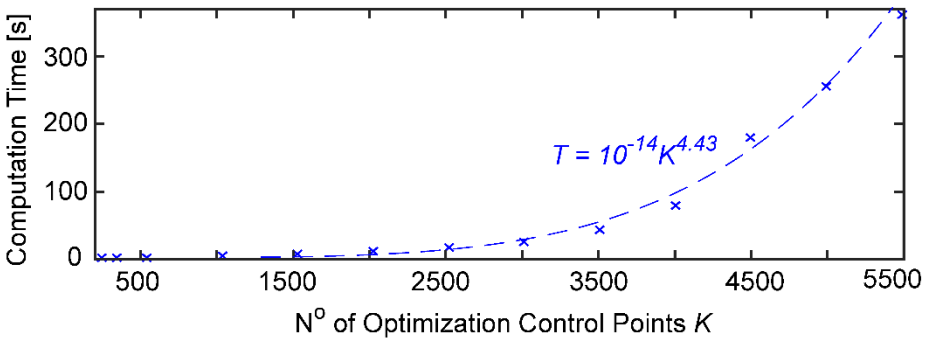

Figure 5-1 Curve fit of computational time for LP

5.2 Toolpath Windowing Algorithm

The proposed new method for handling long toolpaths is based on the Principle of Optimality [47], which implies that a short section taken out of a long trajectory that is optimal, must itself be optimal as well. This is shown in Figure 5-2. First, the entire toolpath is optimized (black), referred to as the 'one-shot' solution. Then, a portion of the same toolpath (red), referred to as the 'windowed' solution, is isolated and optimized with zero boundary conditions (BCs). The resulting feedrate profiles are overlaid for comparison. Although the windowed profile is calculated independently from the rest of the toolpath, there is significant overlap between the windowed and one-shot feedrate profiles.

Considering the feed profile for the windowed tool path Figure 5-2, there are three distinct zones that can be observed. The BC dominant zones (red) at the beginning and end portions of the feedrate profile are shaped according to the enforced zero conditions. After the tool has traveled sufficiently far away from the BCs, the geometry of the toolpath becomes the major factor in influencing the shape of the feedrate profile (green). In the geometry dominant zone, the windowed feedrate profile perfectly overlaps with the one-shot feedrate profile. For any given section taken

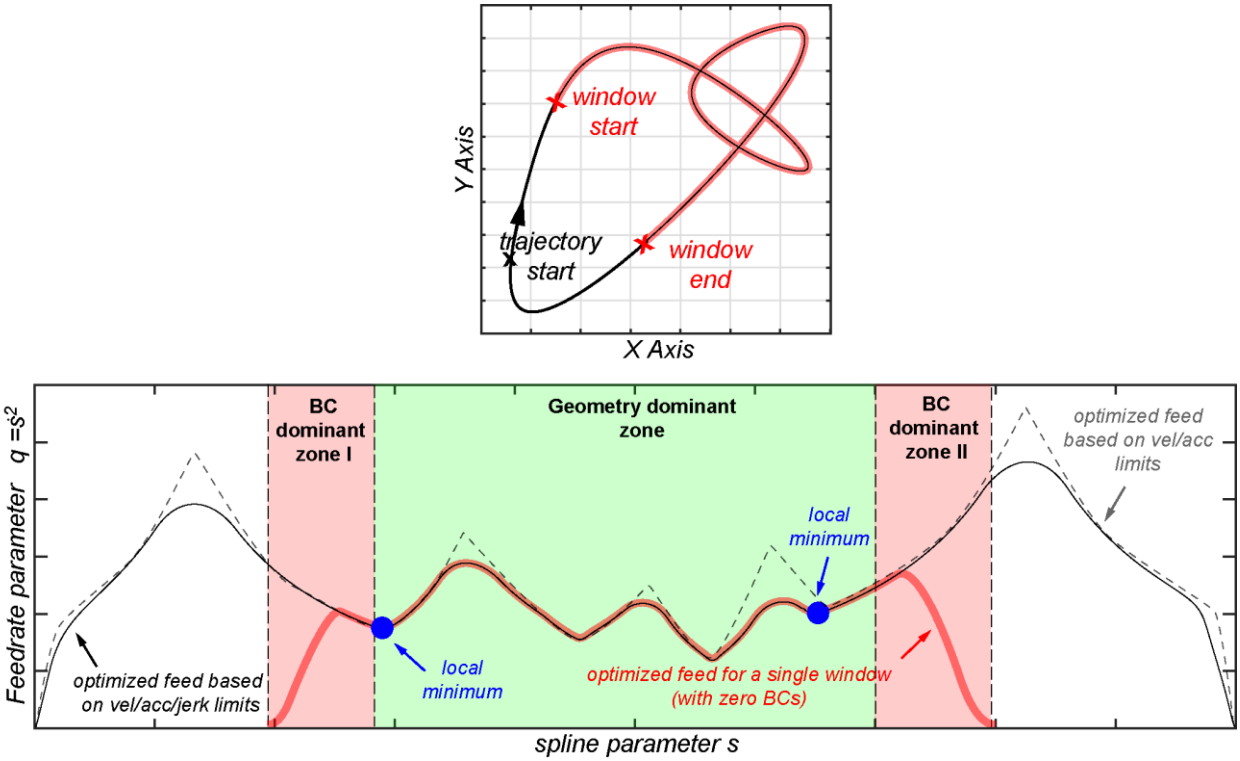

Figure 5-2 Comparison of optimized feedrate profile for two overlapping toolpaths

from the toolpath, the same three zones will exist as long as the window size is longer than the two BC dominant zones combined. The size of the BC dominant zones typically does not extend beyond the first local minimum in the feedrate profile, as can be seen in the example. This is because the minima in the feedrate profile is caused by toolpath geometry, and not the boundary conditions. It is then intuitive that to reduce the computational load of solving a long toolpath, the toolpath can be segmented into smaller windows to be processed independently and then connected to one another to form a continuous feed profile. In this thesis, a novel 3 step parallel windowing (PWin) algorithm has been developed to perform this task, as illustrated in Figure 5-3.

Step 1. Segment the toolpath into windows long enough such that at least one local minima exist in each section. Solve the two stage LP optimization problems described in

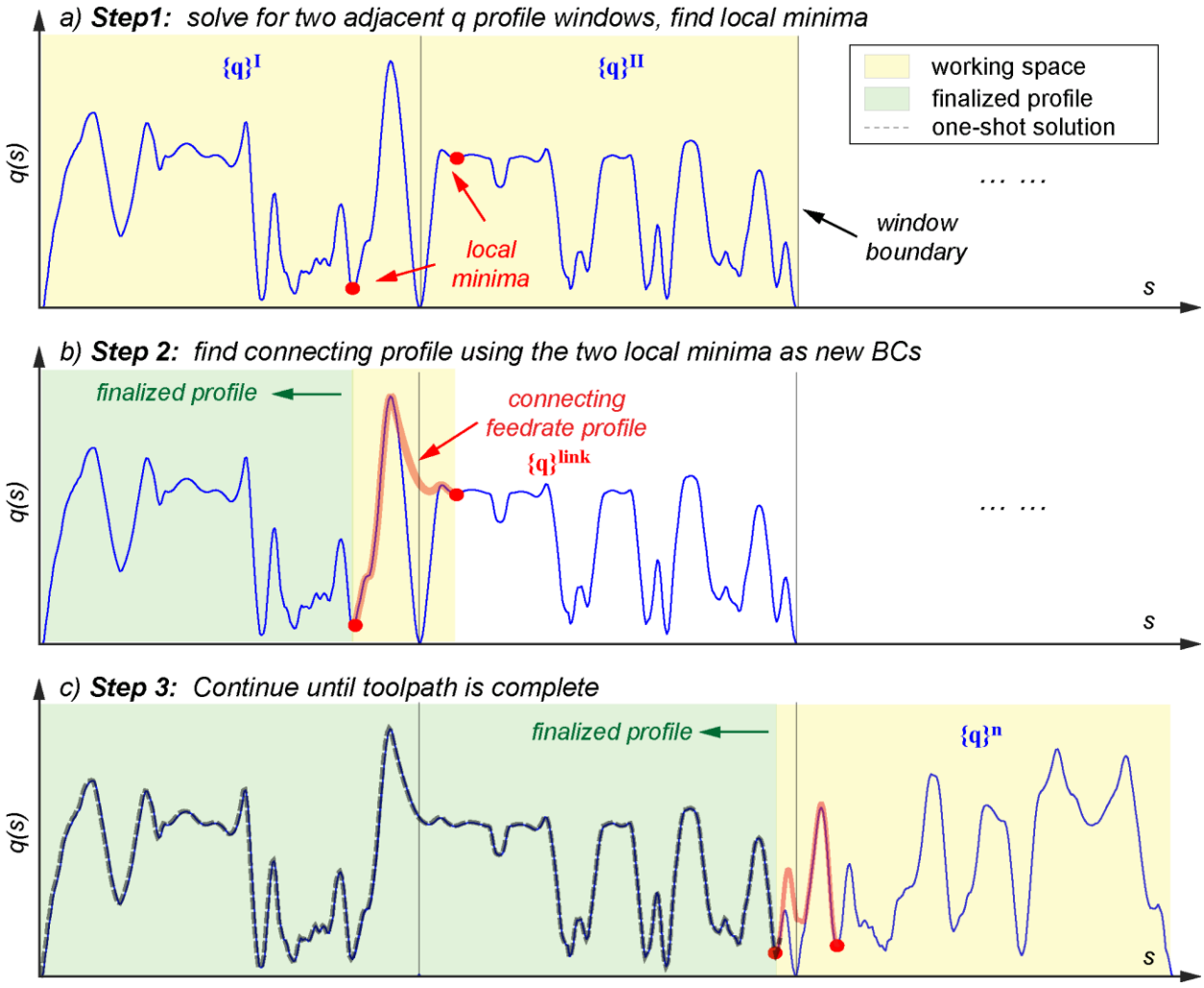

Figure 5-3 Parallel windowing algorithm steps

Chapter 4 for two adjacent windows ($\{q^{I}\},\{q^{II}\}$) with zero boundary conditions ($q' = 0, q'' = 0$, equivalent to $\dot{s} = 0,\ \ddot{s} = 0$). Identify the local minima closest to the shared boundary between the windows, which are to be used as connection points.

Step 2. Determine q' and q'' for the local minima found in the previous step. In the optimization problem, these become the BCs for the $\{q^{link}\}$ profile, to ensure that the velocity and acceleration are continuous at the connection. Feasibility is guaranteed, since there already exists a feasible profile between these two points which passes though $q' = 0, q'' = 0$ from the previous step.

Step 3. Solve the windowed profile with zero BCs for the next window and repeat the process from Step 2 until the entire toolpath is processed.

Implementation of Steps 1 and 2 within themselves are parallelizable, as opposed to the sequential processing used in mainstream feed optimizers. Computations of individual windows and individual connecting profiles are independent. Such parallelization can also be benefit future super high-speed machine tools, by keeping the computational time of very fast trajectories below their actual execution time.

Since one of the major features of the algorithm, which makes long toolpath processing efficient, is the use of local minima as connections between feedrate profile segments, the initial selection of window size during Step 1 is important. If the window size is too small, the solution will be sub-optimal, since the profile will not have enough travel distance to reach the first local minimum. Axial limits of the CNC machine can be used to calculate the minimum window size to guarantee that at least one local minimum will exist within a window for a specific machine. First,

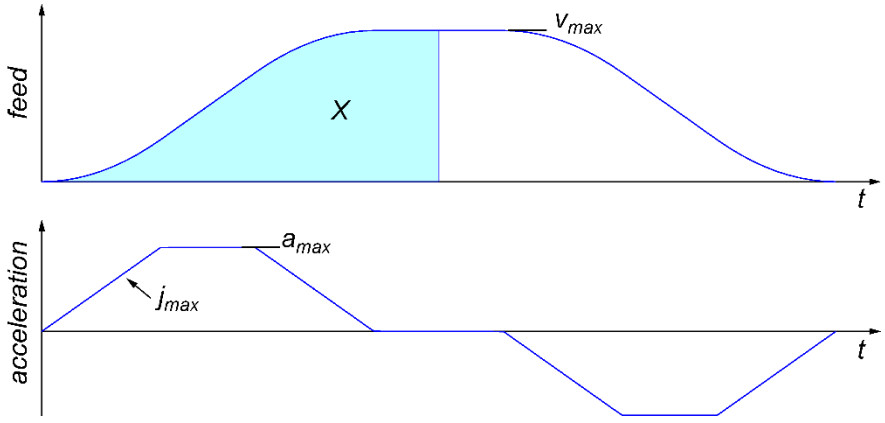

Figure 5-4 Characteristic distance X for given CNC kinematic limits

a feed profile is constructed in the time domain, such that its acceleration and jerk profiles are maximized. Then, the area under the feedrate curve will be half the minimum displacement (X in Figure 5-4) required for the feedrate to contain either a local minimum or plateau. Afterwards, the toolpath can be parsed before applying feedrate optimization. Typically, the window size is chosen to be at least two to three times of X.

In cases where a window size adequately long cannot be used, due to e.g. memory/storage limitations, the proposed algorithm can also be implemented iteratively, to improve the optimality of the final solution. In Figure 5-5a for example, the initial windows are purposely chosen to be too small (dark gray). It can be seen that most windows do not touch the optimal one-shot solution (red dashed). An extra step, Figure 5-5c, is added after Step 2, where instead of using zero BCs, each window is re-calculated using updated BCs taken from the connection profiles. Then, steps b and c are iterated until the solution converges. Figure 5-5d shows the final solution after 5 iterations. There are still locations where there is some sub-optimality. This is partly caused by the constraint tightening applied for numerical robustness, which is discussed in Section 5.5. Nevertheless, as

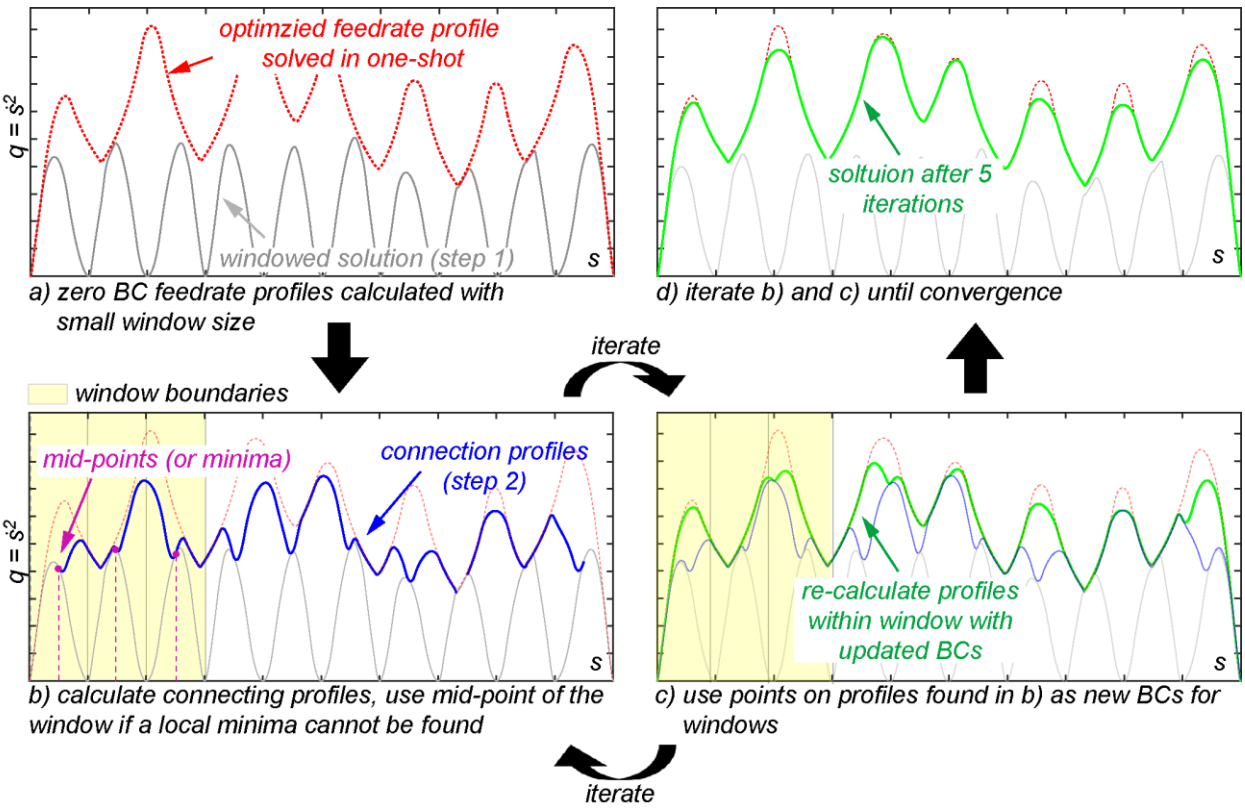

Figure 5-5 Possible algorithm iteration for implementation where window size has to be kept short

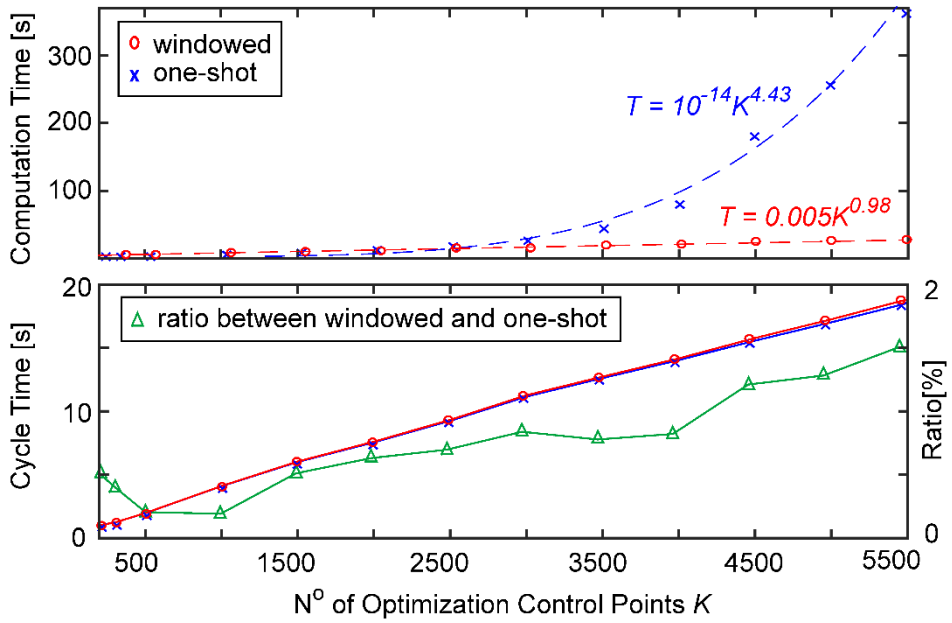

Figure 5-6 Computational load and optimized trajectory cycle time comparison between one-shot and LP+PWin algorithms

this example demonstrates, iterations can be used to further enhance the optimality of the proposed method, when the window size has to be kept limited.

An important feature of the proposed windowing algorithm is that the calculation of the feedrate profile in each window in Step 1 is entirely independent, and the connection profile is only governed by the two neighbouring windows. This not only reduces the computational time, which normally grows by polynomial order of 4.4 to linear complexity, as shown in Figure 5-6, but also provides other advantages as well. Some of these advantages are easy implementation of data streaming for long toolpaths, real-time toolpath processing, and the capability to use parallel computing. As a trade-off in return for the achieved significant reduction in the computational time, the loss of optimality for the solved trajectory is only around 1-2%. This is believed to originate from robustness measures taken to reduce the effect of numerical errors, described in Section 5.5.

5.3 Sequential Quadratic Programming (SQP)

To further improve the optimization of the feedrate profile, it is better to use the real jerk constraint rather than the estimated pseudo-jerk. One method of incorporating non-linear constraints is to use an SQP solver. SQP solvers have the disadvantage of computational inefficiency when the initial guess of the solution is not given, or the guess is too dissimilar from the optimized solution. However, the fast LP+Pwin algorithm is able to provide a good close guess

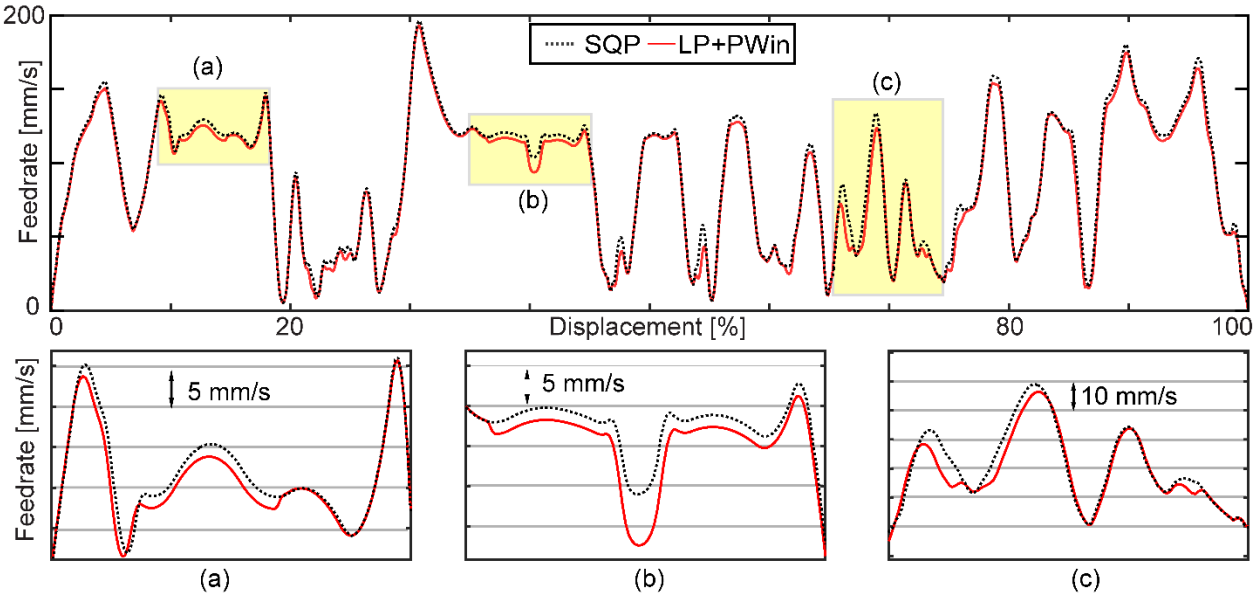

Figure 5-7 Comparison between LP+PWin and SQP solutions

for the optimized solution. Thus, using the estimate obtained from the LP+Pwin algorithm as the initial condition for the SQP solver can decrease the amount of time required for convergence. The advantage of adding the SQP step after performing the LP+Pwin is further optimization of cycle time. For repeating processes that do not require real-time processing, where the goal is to keep the cycle time as short as possible, SQP based feedrate optimization could provide a further 2~5% improvement in cycle time over the LP solution.

5.4 Time Domain Reconstruction

To linearize the optimization problem and find explicit expressions for the required derivatives, all operations and calculations are performed in the arc displacement, s domain. To obtain the feedrate profile in time (t) domain, so that the generation of a time-functioned trajectory which can be interpolated by the CNC is possible, the following integral must be evaluated:

$$Total\ Time = \int_0^T dt = \int_0^1 \frac{dt}{ds} ds = \int_0^1 \frac{1}{\dot{s}} ds \tag{5.1}$$

By substituting the definition of $q = \dot{s}^2$ into Eq. (5.1), the integral can be evaluated in terms of q and s:

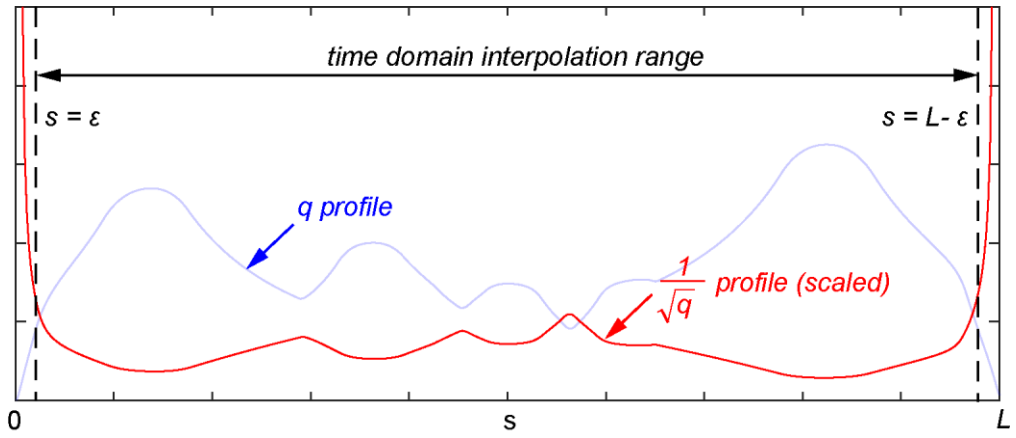

Figure 5-8 $1/\sqrt{q}$ **profile trimming for time domain interpolation**

$$t_i = \int_{s_{i-1}}^{s_i} \frac{1}{\sqrt{q}} ds \tag{5.2}$$

Under the assumption that there exists a function which directly relates *t* and *s*, Taylor series expansion ignoring the higher order terms, has been used to find the corresponding s value for each time-step as shown in Eq. (5.3). The size of the time-step is chosen based on the required trajectory interpolation sampling frequency of the CNC machine.

$$s(t_i + 1) = s(t_i) + \dot{s}(t_i)dt + \frac{\ddot{s}(t_i)}{2}dt^2 + \frac{\dddot{s}(t_i)}{6}dt^3 + H.O.T. \tag{5.3}$$

Where:

$$\left.\begin{aligned} \dot{s} &= \sqrt{q} \\ \ddot{s} &= \frac{d\dot{s}}{dt} = \frac{d\dot{s}}{ds}\frac{ds}{dt} = \dot{s}'\dot{s} \\ \dddot{s} &= \frac{d\ddot{s}}{dt} = \frac{d\ddot{s}}{ds}\frac{ds}{dt} = \frac{d(\dot{s}'\dot{s})}{ds}\dot{s} = [\dot{s}''\dot{s} + (\dot{s}')^2]\dot{s} \end{aligned}\right\} \tag{5.4}$$

$\dot{s}'$ and $\dot{s}''$ are obtained through differentiation of the $\dot{s} = \sqrt{q}$ profile, by fitting a cubic spline to $\dot{s}$ as a function of *s*. Experimentally, it is shown that the cubic spline fit will not introduce significant inaccuracies in the time domain interpolation. For stop conditions, where the CNC starts from or arrives at rest, the proposed algorithm constrains the beginning and end of the $q(s)$ profile

to be zero. However, when evaluating the integral $t_i = \int_{s_{i-1}}^{s_i} \frac{1}{\sqrt{q}} ds$, $\frac{1}{\sqrt{q}}$ approaches infinity at these locations, as shown in Figure 5-8. The Taylor series expansion will stall, since evaluating the area under an unbounded curve results in infinite motion time estimation. Therefore, instead of beginning the interpolation at $s = 0$ and ending at $s = L$, a small offset is applied ($s = \varepsilon$), in order to make the integral finite. The offset will affect how long the trajectory will take to start and stop. The optimal offset is typically found by trial and error, since different machine tools may be capable of absorbing different amounts of discontinuity; if the trajectory is interpolated at a sampling period of Ts, using an offset of ε would result in a sudden velocity command of ε/Ts, a tangential acceleration command of ε/Ts^2, and a tangential jerk command of ε/Ts^3.

5.5 Compression of Feed Drive Limits for Numerical Robustness

In the proposed algorithm, recall that in Step 2, a feasible solution can always be found, since the feedrate profiles from Step 1 are feasible. However, numerical errors are present in LP solvers in practice. For example, it is possible that the solution found in Step 1 may slightly exceed the

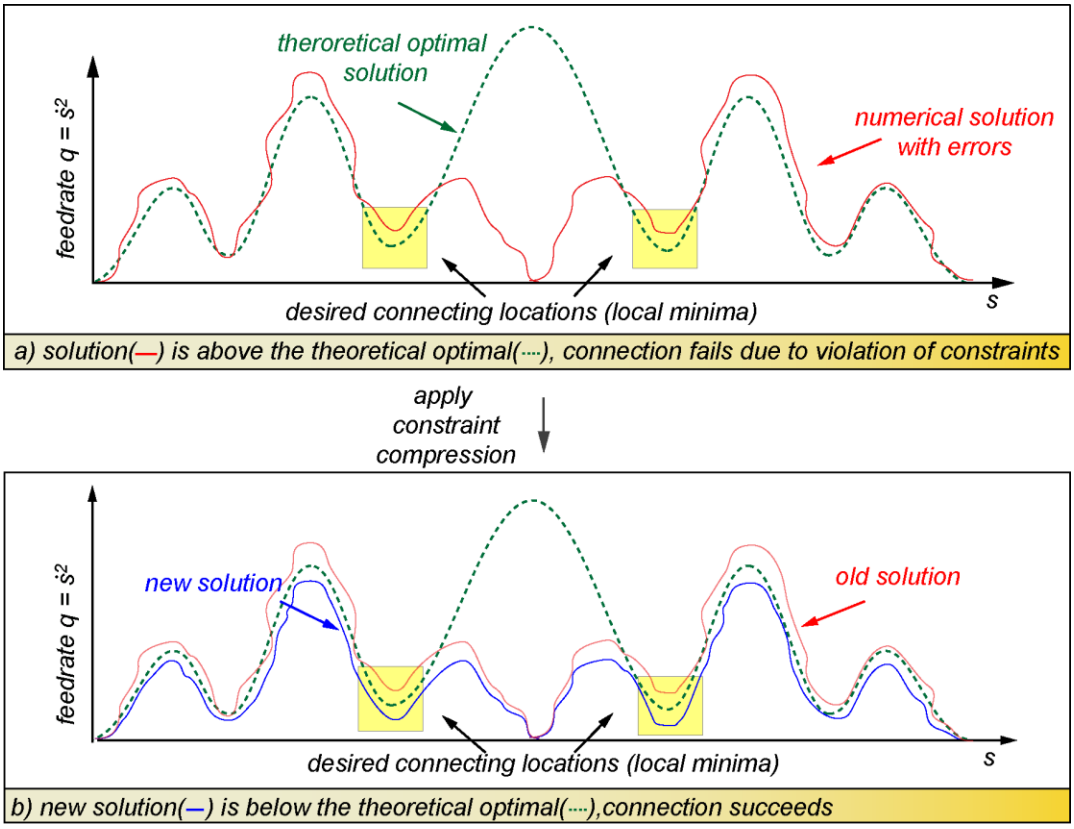

Figure 5-9 Constraint compression to improve robustness

upper limit for velocity, acceleration and/or jerk due to rounding, as illustrated in Figure 5-9a. When points taken out of such feed profiles are used as boundary condition equalities in Step 2, this may cause errors in the solver since they are points that do not lie on a feasible profile. To resolve this problem, a compression factor is applied to the limits used in Step 1 of the algorithm such that q is calculated using reduced limits: $[\alpha \cdot v_{\max,}\ \alpha \cdot a_{\max}, \alpha \cdot j_{\max}]$, where α is less than 1 (usually between 0.90 to 0.99). The tightened constraints force the calculated q profile to be lower than the optimal (Figure 5-9b), reducing the likelihood of any minor constraint violations that might get passed onto Step 2.

Chapter 6
Implementation and Results

6.1 Introduction

The proposed LP+PWin algorithm has been fully tested in simulation and experimental results at the University of Waterloo and in industry at Pratt and Whitney Canada (P&WC). In this chapter, Section 6.2 compares the performance of the proposed LP+PWin algorithm against the Forward Projection [36] method and VPOp [35] method, referred to in Chapter 2. Section 6.3 verifies the robustness of the proposed algorithm considering a complex 3D surface. Section 6.4 and Section 6.5 present the results obtained at P&WC on two different machining operations, that have been identified to be able to benefit from feedrate planning.

6.2 Benchmark with other methods

A 2D profile in the shape of a cat has been prepared for testing on a three axis flatbed CNC router with ball screw drives, shown in Figure 6-1. The same wood router is used for robustness verification in 3D machining tests as well (Section 6.3). The router's drives have the following kinematic limits:

Figure 6-1 Experimental setup (4'x8' wood router table)

Table 6-1 Flatbed router table axis level kinematic limits

	X axis	Y axis	Z axis
Velocity [mm/s]	150	150	150
Acceleration [mm/s^2]	500	500	500
Jerk [mm/s^3]	10,000	10,000	10,000

For the proposed LP+PWin algorithm, the optimized feedrate profile for the cat shaped toolpath is obtained by dividing the toolpath into 10 windows, and considering 100 constraint evaluation points in each window. The code is executed using a Core i5 2.6GHz processor with Windows 10 environment. The resulting feedrate profile is compared against two other previously developed feed optimization algorithms: Forward Projection [36] and VPOp [35]. The tangential feedrate profiles and axis level kinematic profiles optimized from the three different algorithms are presented in Figure 6-2. The contouring cycle time and required computation time comparisons

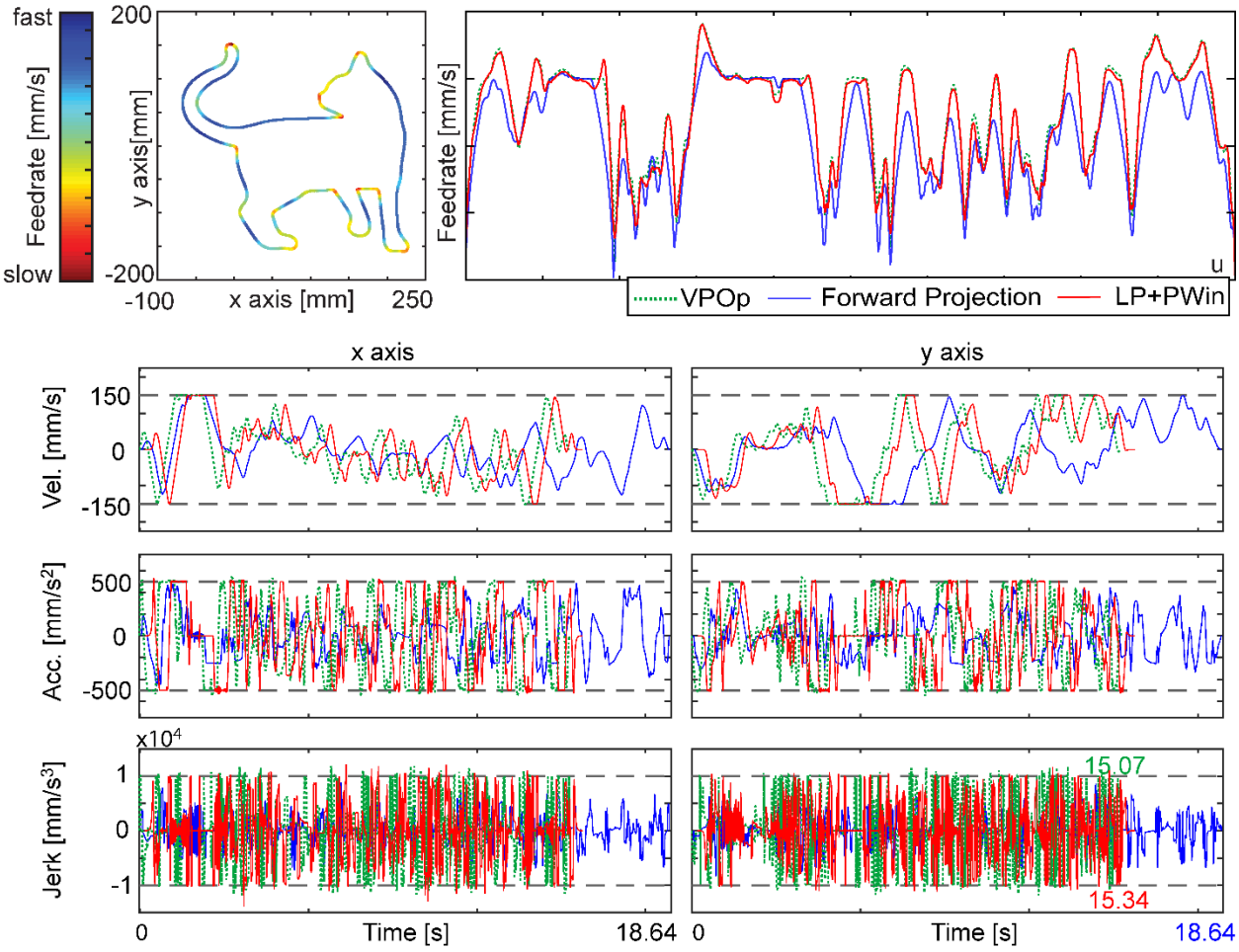

Figure 6-2 LP+PWin 2D optimization result comparison with Forward Projection and VPOp

are summarized in Table 6-2. In the table, results from a SQP formulation using non-linear constraints, as described in Section 5.3, are also presented. The SQP result is able to achieve the shortest motion time, and has been considered as a benchmark for optimality. Due to the similarity in the feedrate profiles between the SQP, VPOp, and LP+PWin, the feedrate and kinematic profiles for SQP are not shown in Figure 6-2, for the sake of clarity. As expected, the computation time for SQP is extremely long, at over 5 hours for a relatively simple toolpath.

From Figure 6-2 and Table 6-2, it can be concluded that the LP+PWin algorithm has the advantage of finding a result close to the optimal one while keeping computational time low. The 3% sub-optimality in the cycle time reduction, when compared to the VPOp method, is partly due to the constraint compression described in Section 5.5 to ensure numerical robustness. From the axis level kinematic profiles, it can also be seen that another source for increased cycle time is the evaluation of the inverse feedrate integral in the time-domain interpolation. Where at the beginning and end of the trajectory, the feed is slow to accelerate and decelerate from and to zero, due to the small feedrates considered.

Table 6-2 Cycle time reduction and computational time comparison for the cat toolpath

	Cycle Time	Computation Time	Cycle time increase
VPOp	15.07s	>600s	1%
Forward projection	18.64s	3.12s	25%
Proposed LP+PWin	15.34s	10.39s	3%
SQP	14.90	>5 hours	fastest

6.3 Robustness evaluation in 3D surface machining experiment

To test the robustness of the algorithm, a 3D wavy surface (an enlarged molar tooth) has been prepared. In this case, 100 windows are used with 100 constraint evaluation points within each window (10,000 constraints considered in total). For the two step LP algorithm, 9 inequality equations must be solved for each constraint point in each axis, as shown in Table 6-3. Therefore, in this particular experiment, disregarding the equality constraints and connecting profile re-calculations, a minimum of 270,000 constraints are evaluated in 62.48s, where the Forward Projection algorithm takes 31.58 seconds.

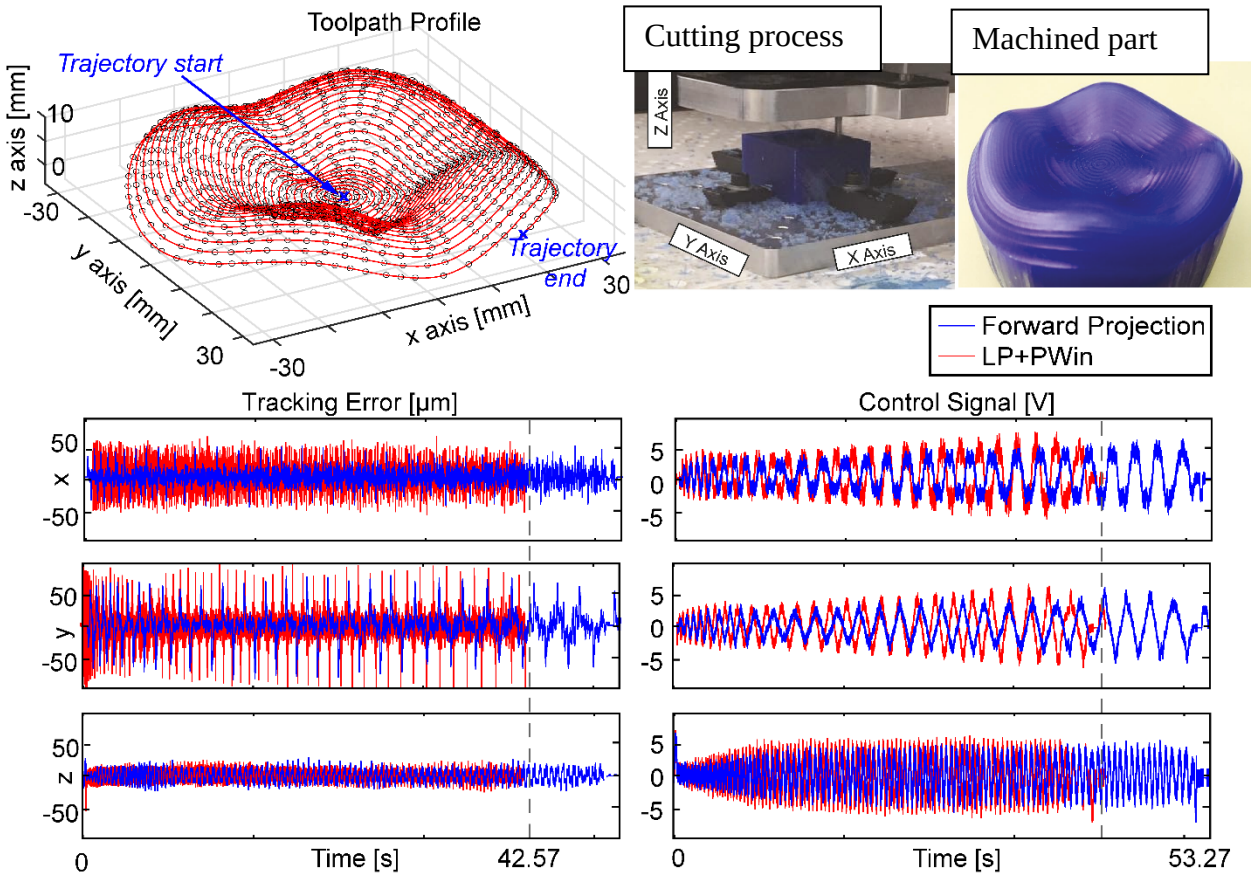

Figure 6-3 3D optimization result comparison

Table 6-3 Number of inequalities for each constraint evaluation point

Optimization of q^*	Velocity	→ 1 inequality
	Acceleration	→ 2 inequalities
	Jerk	→ 1 inequalities
Optimization of q	Velocity	→ 1 inequality
	Acceleration	→ 2 inequalities
	Jerk	→ 2 inequalities

The tooth shaped part is machined from a soft plastic material using the router with kinematic limits as presented Table 6-1. The tracking error and control signals are recorded from the machine, and shown in Figure 6-3. Also shown in the same figure are a snapshot from the cutting process and a picture of the machined part. It can be seen that the proposed algorithm is able to reduce the cycle time by approximately 20% compared to the Forward Projection method, while keeping the

tracking error within similar magnitudes. There is no comparison to the VPOp or SQP, since these algorithms are unable to successfully produce a feedrate profile for the toolpath in measureable time, due to the large computational load. By evaluating the axial velocity, acceleration and jerk profiles of the resulting optimized trajectory, it can be seen that none of the given limits are exceeded, as shown in Figure 6-4.

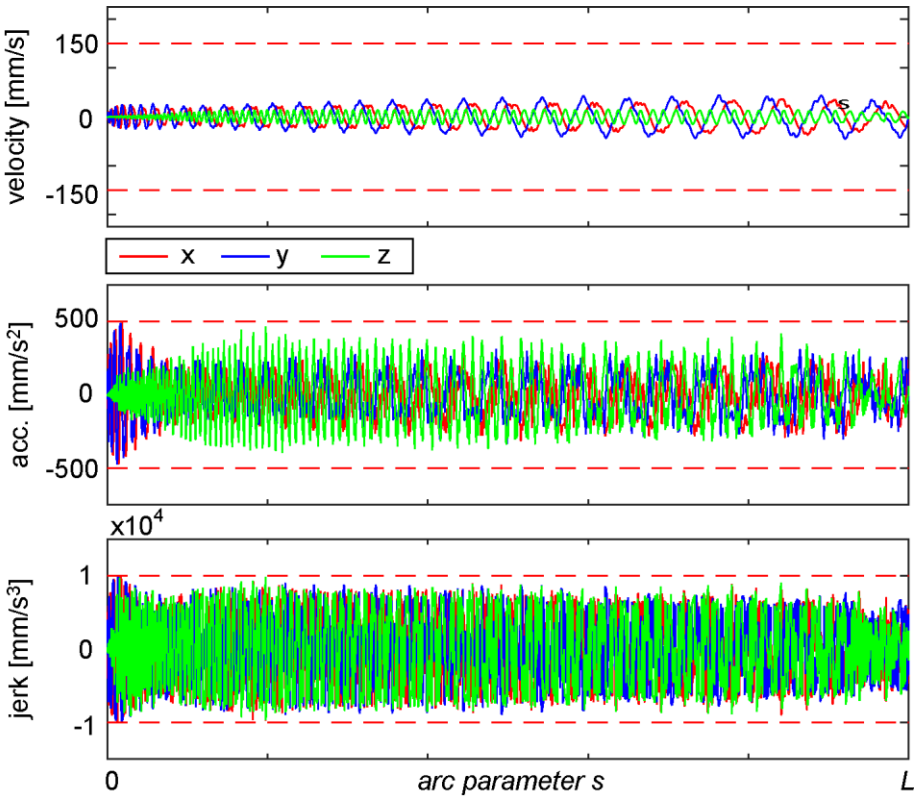

Figure 6-4 Velocity, acceleration and jerk profiles for the tooth toolpath

6.4 Industrial implementation #1, firtree machining for turbine disks

The proposed LP+PWin algorithm has been verified on a general firtree profile contouring operation, for jet engine turbine disks at P&WC. The contour is named as such for its similarity to a firtree shape, which is used to lock the turbine blades onto a turbine disk, so the two parts do not

Figure 6-5 Descriptive images for the test machine and sample part [48]

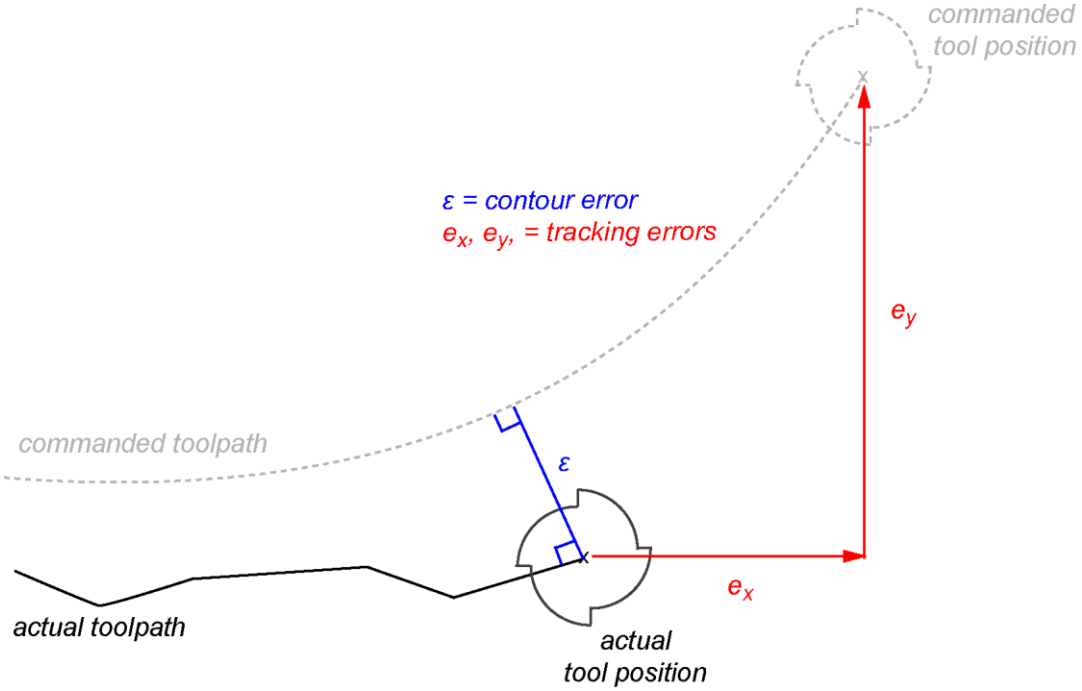

Figure 6-6 Tracking and contour error definitions for evaluating test results

become detached from centripetal force while the turbine engine is in operation. Due to company confidentiality reasons, the actual machine tool and part are not shown in this thesis. A similar machine tool and an example part, from images available in the public domain, are shown in Figure 6-5. All results presented in the later figures have their critical dimensions and cycle times redacted, in line with the industry sponsor's clearance policy regarding academic publications.

The experimental goal is to process the G-codes provided by Pratt & Witney Canada (P&WC) using the cubic spline method, discussed in Chapter 3, and optimize the process according to the LP+PWin formulation, described in Chapter 4. The test part is manufactured using a 5 axis Mitsui Seiki Vertex machine. The results are analyzed based on two criteria shown in Figure 6-6: tracking error, defined as the distance between the current tool position and desired tool position; and contour error, defined as the minimum distance between the originally defined toolpath in P&WC's unmodified NC code, and the instantaneous actual tool position. Tracking errors can be directly recorded from the CNC, through monitoring of the command and encoder signals. Contour errors are calculated from comparing the encoder data to the original commanded G-code comprising linear and circular arc movements, as described in Chapter 3, based on P&WC's original part program.

Different from the traditional G-code format described in Chapter 3, the optimized trajectory is passed to the machine using a special built-in function of the Fanuc controller, called 'inverse-time mode'. In inverse-time mode, a series of locations and frequencies (units of inverse minute) are passed to the controller, as shown in the example in Figure 6-7. The commanded axis will move

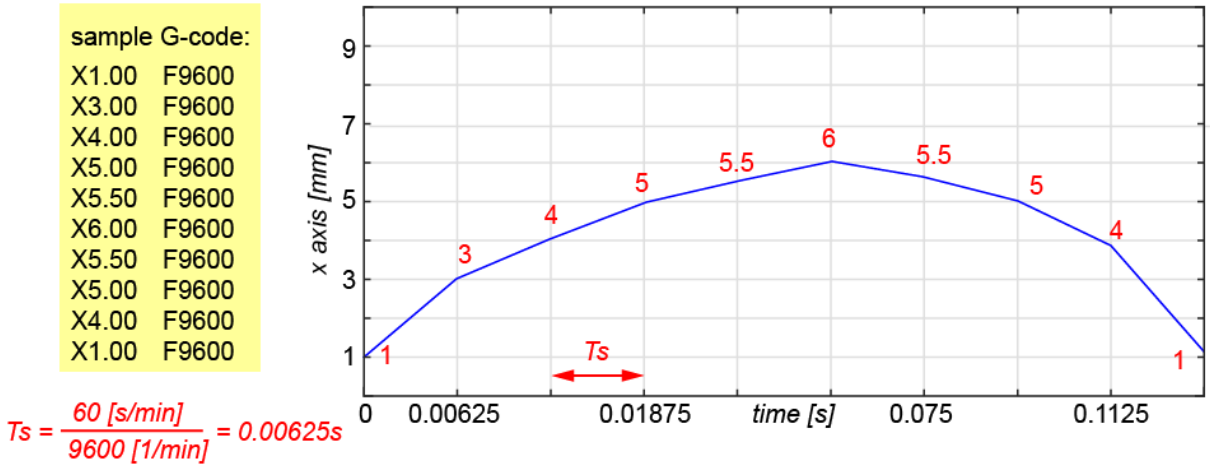

Figure 6-7 Example of inverse-time mode G-code

between the given points according to the sample time (*Ts*), calculated from the feed values given by F. The waypoint frequency may vary between point to point, or remain the same. This gives the capability to machine complex shapes that are not easily defined by lines and circles, using time domain trajectories which can be generated off-line, with reasonably high temporal resolution.

First, response of the CNC during the original fir tree machining process is carefully studied. After reading data from the current process, it is found that due to the nature of the G-code, and tolerance settings specified in the CNC, after each linear or circular interpolation, the machine briefly stops, as seen in Figure 6-8, curve (a). This behaviour is not desirable, as each time the machine's drives go through a zero velocity condition, the increase in static friction caused a sudden spike tracking error profile, shown by the green curve at the bottom panels of Figure 6-8 . These spikes will often leave physical feed marks on the part, and the unnecessary drop in feedrate will adversely affect cycle time. From recorded data, it is also found that the current feed for each section is much lower than the capabilities of the machine. These feed values are pre-determined by P&WC process engineers, based on other physical factors such as cutting forces, material removal rate, and tool-life considerations. Therefore, for consistency, these feed limits are retained during the first attempt of optimization, as shown in Figure 6-8 curve (b).

The current kinematic limits of the operation (axis level velocity, acceleration, and jerk) are found by reading the position data from the machine, then filtering and differentiating the results to obtain the maximum kinematic values. Using these limits, optimization is performed according to the LP+PWin algorithm. This helped eliminate the start and stop behaviour completely. In this case, the motion time could be reduced by 14% without significantly affecting the current level of contour error (~5μm). Additionally, the tracking performance could be improved with the optimized trajectory, by the removal of the spikes at segment boundaries.

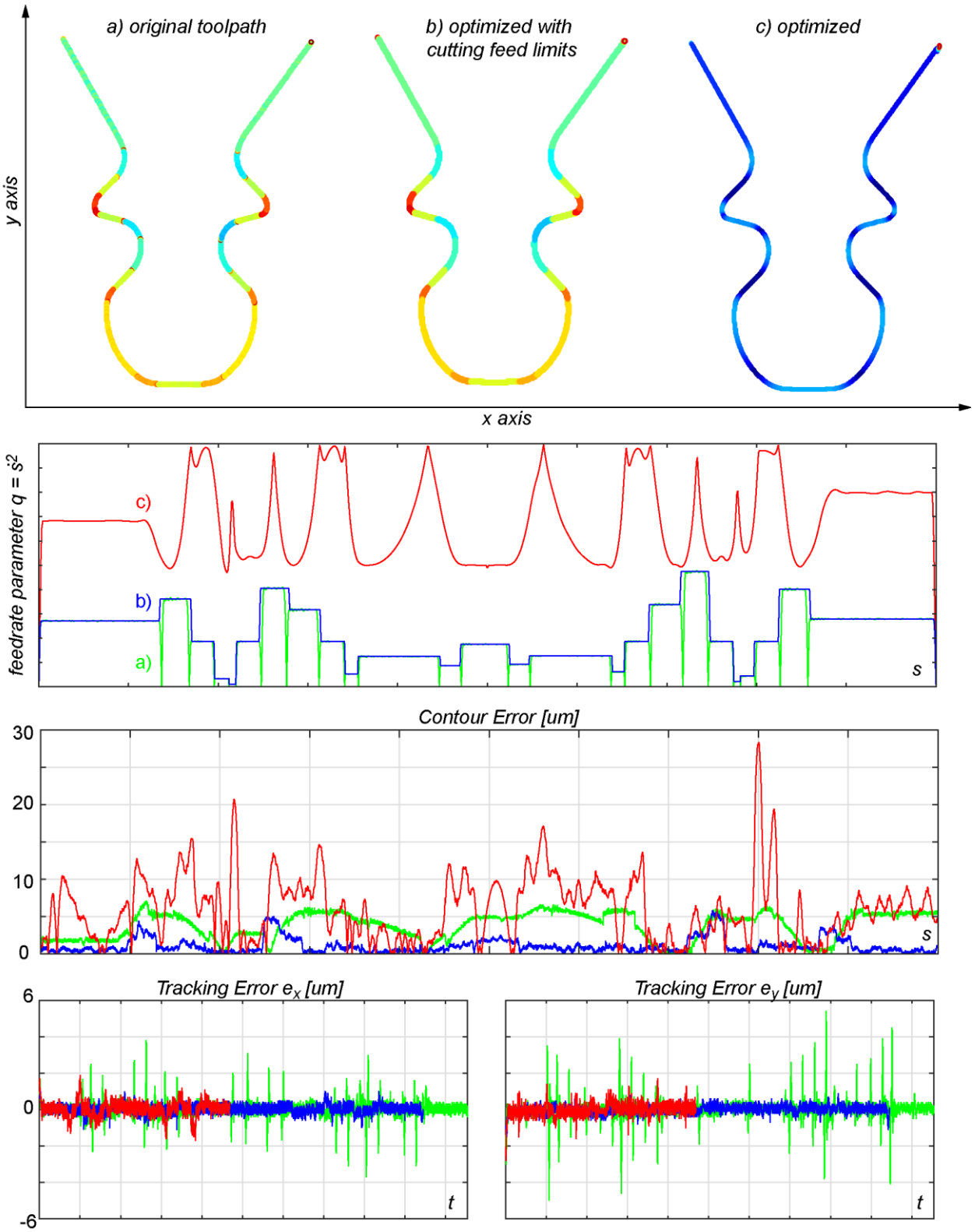

Figure 6-8 Fir-tree optimization results

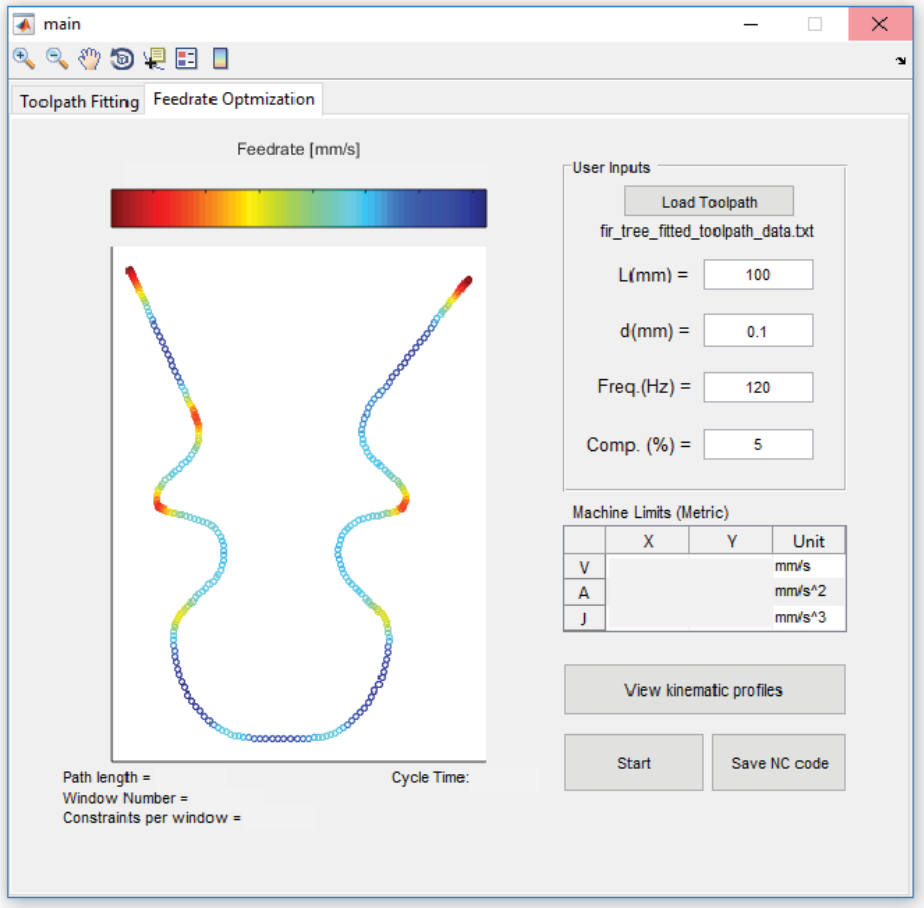

Figure 6-9 Firtree toolpath feed optimization module

In the second optimization attempt, the cutting limits are removed such that the profile is optimized considering only the CNC machine kinematic capabilities, as shown in Figure 6-8 curve (c). This cycle time reduction is 57% in this case. However, due to the increased speeds, it can be seen that the contouring performance is sacrificed. Investigation into collected data reveals that the increase of contour error may be caused by unknown functions inside the CNC's interpretation of inverse-time commands, which slightly altered the commanded trajectory with increasing velocity. This is evident in the fact that the tracking performance is not compromised with the increase in feed. Further investigation is required, in collaboration with Fanuc, to identify the root cause and disable this effect inside the controller.

It is planned that additional studies will be conducted by P&WC process engineers using the developed Graphical User Interface (GUI) for the process, as shown in Figure 6-9, to find a suitable compromise between cycle time and contouring performance for future process developments.

6.5 Industrial implementation #2, air foil blade machining

Another operation with the potential for feedrate optimization is the contouring of air foil blades on a turbine disk similar in shape to the ones shown in Figure 6-10. As the means of material removal, a spinning torus shaped grinding disk is attached to the end of the spindle, moving along the surface of the blade to grind away material. This process is referred to as Super Abrasive

Figure 6-10 Turbine air foil blades [49]

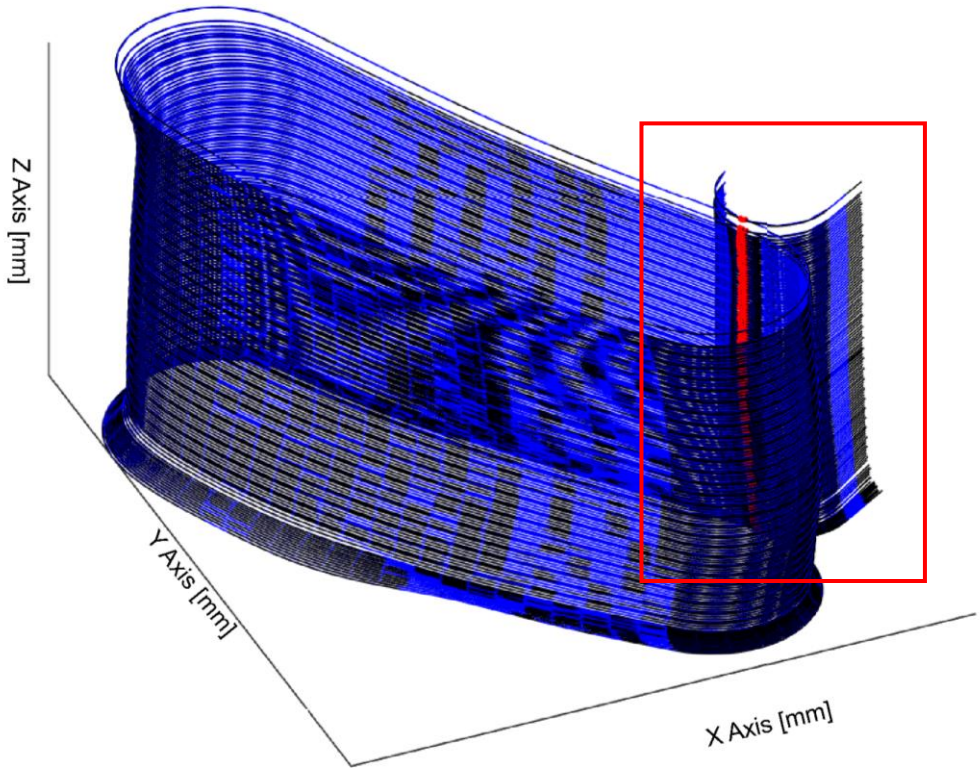

Figure 6-11 Single air foil blade toolpath

Machining (SAM). The full toolpath for a single blade is shown in Figure 6-11. About 20% of the passes are for roughing and the rest are for finishing. The major difference between the two types of passes is the type of grinding disk used, with no significant variation in toolpath shape or feed. During tool engagement with the workpiece, there is only movement in the XY plane (2D motion only). The tool enters and exits the workpiece (red) with a circular path to achieve the desired lead in and out of the work-piece, and then moves down in Z with rapid motion (G00) to the next pass. Since the tool is torus shaped, and significantly wider than the spindle shaft, the shape of the fan blade can have minor undercuts, apparent at the top and bottom of the blade, as shown in Figure 6-11. These features can be produced without requiring movement in the rotary axes. A high number of passes are required to produce a single blade. Therefore, it is evident that to machine a complete turbine disk with a large number of blades, the process cycle time can become very lengthy. Thus, it is important to apply feed optimization to the SAM process with efficient contouring, in order to reduce the cycle time.

Different from the firtree operation in the previous section, the toolpath in the original process for the air foil is programmed with a single constant feedrate for each complete pass. Figure 6-12 shows the optimization result for a single pass of an air foil blade viewed in the XY plane. As expected, the start and stop behaviour present in the fir-tree is not present in the SAM trajectory, and the feed is constant along the toolpath, as can be seen in Figure 6-12 curve (a). After optimization, it can be seen that the motion time is reduced by approximately 40%, and since both trajectories are produced by the CNC using the same form of toolpath smoothening and pre-filtering, the deterioration in the contour error is very similar to that of the original process. Additionally, the tracking performance has remained the very similar, as shown in the bottom panels of Figure 6-12. Following the same procedure, all passes have been optimized and the results are summarized in Table 6-4, it is verified that all passes produced similar tracking and contour errors.

Table 6-4 Cycle time reduction for SAM fan-blade

	Cycle time reduction
Roughing Passes	38.06%
Finishing Passes	39.05%
Total	38.81%

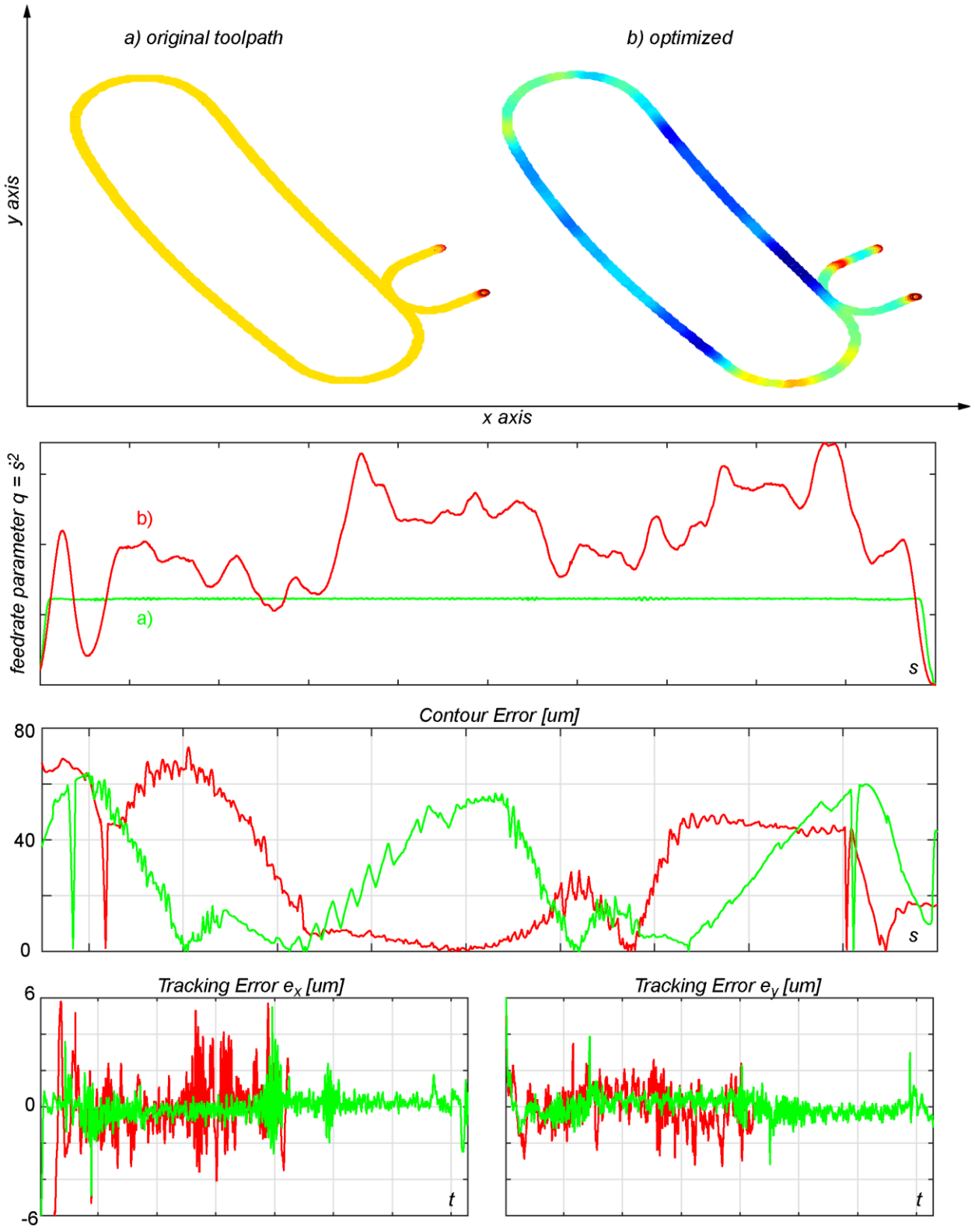

Figure 6-12 Fan-blade optimization result comparison

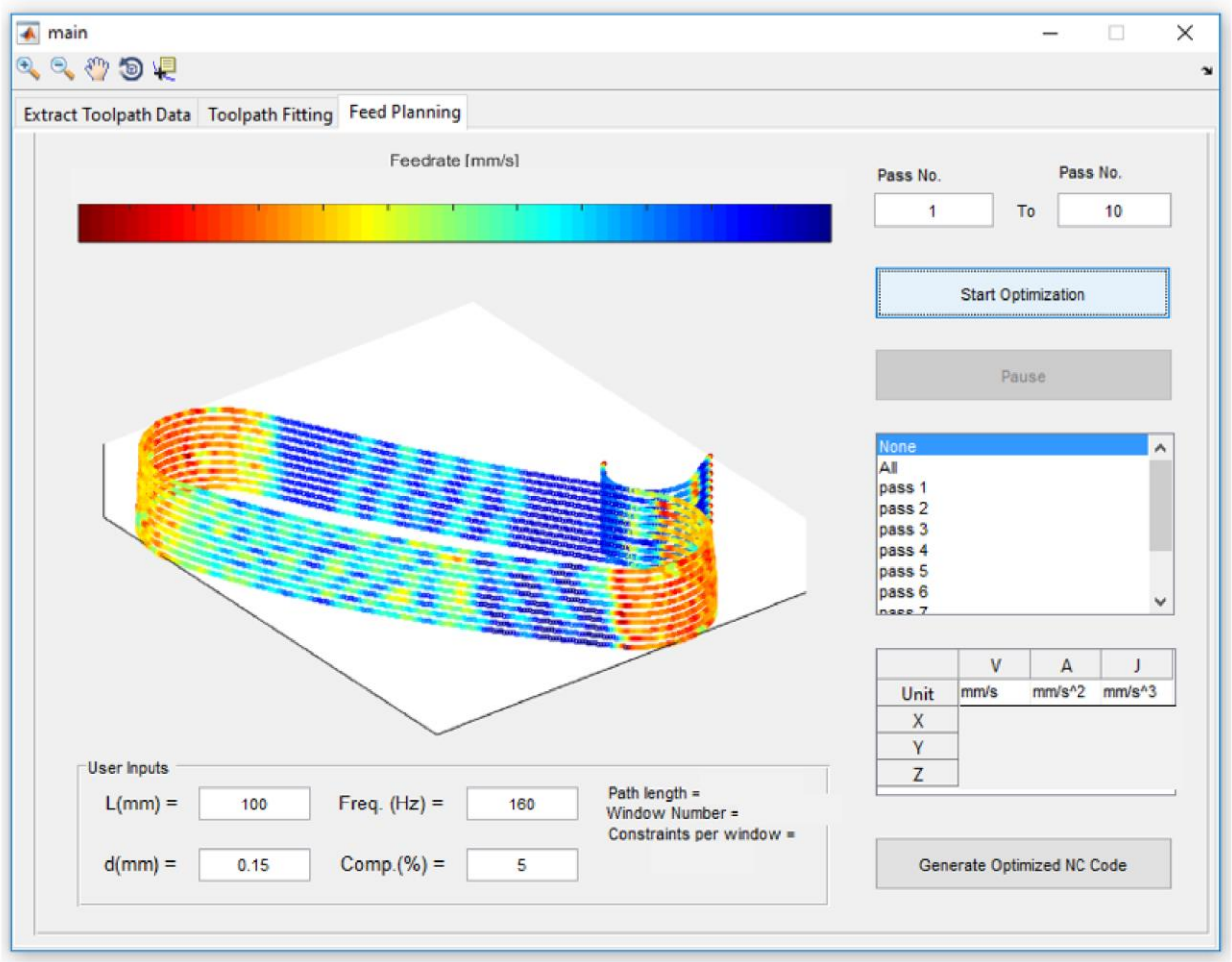

Figure 6-13 SAM GUI toolpath feed optimization module

For the SAM process, each pass is slightly different in shape, and therefore need to be considered separately. The movement between each XY plane is rapid motion (G00), used for tool positioning without contacting the workpiece, G00 can operate at a much higher level velocity, acceleration, and jerk than what is allowed in actual machining. In the current implementation, the rapid movements have been kept as is, with the intention of blending them into an optimized path in future development. A special GUI is developed for this process that has the same features as the firtree interface. Additionally, the process planner is able to select each path individually and optimize. A listbox is added to allow this, shown in Figure 6-13.

6.6 Conclusions

In this chapter, it has been shown that the proposed LP+PWin algorithm performs extremely well, in terms of cycle time reduction and computation time, against the two reported Forward Projection [36] method and VPOp [35]. The algorithm is robust, guaranteed to produce a feasible feedrate profile that is close to optimal for toolpaths of any length and complexity. For industry machining experiments, the proposed algorithm has been shown to be able to reliably improve the productivity of the processes tested by 15~40%. The GUIs developed during this research will aid the process designers at P&WC in further evaluating the proposed algorithm for machining process optimization.

Chapter 7
Conclusions and Future Work

This thesis has presented a method of feedrate planning for long toolpaths. A successful cubic spline fitting algorithm has been developed for processing G-Codes. The iterative fitting method enables the smoothening of toolpaths within a set tolerance, later to be used in feedrate planning. The axis level velocity, acceleration and jerk limited feedrate optimization problem is linearized by a combination of the B-spline formulation and pseudo-jerk approximation, allowing for the utilization of fast and robust Linear Programming optimization. Developed considering the Principle of Optimality, the novel PWin algorithm splits the toolpath into small workable segments, then joins them at strategic points. This gives the algorithm the freedom to incorporate parallel computing and streaming capabilities, which are suitable for use in next-generation high speed machine tools, without significantly compromising the optimality and convergence time of the results. The computational time is also shown to have been reduced to a linear growth rather than a polynomial growth of order 4.4, compared to solving the complete trajectory in one shot

The LP+Pwin algorithm has the capacity to handle indefinitely long toolpaths in real time, and produces feasible feedrates that respects any given axis level or tangential limits from the machining process. Compared to mainstream methods, the proposed algorithm is able to reliably produce a feedrate profile that close to the optimal solution calculated by the non-linear solvers, while keeping the computational time on a similar level to a much simpler and faster heuristic algorithm.

Simulation results show that LP+PWin is robust in handling a large amount of constraints. For one test, the algorithm has been shown to be able to continuously process over 270,000 constraint equations successfully without stalling. Physical experiments performed at the UW PCL show that the algorithm is able to produce feedrate profiles that reduce the cycle time by approximately 20%, over heuristic (forward projection based) look-ahead, without sacrificing the tracking performance and computational efficiency. Industrial experiments conducted on two separate processes at P&WC show that the proposed LP+PWin algorithm is able to consistently reduce the motion cycle time by anywhere between 15% to 40%.

The results from this research have been packaged into workable code, with a fully functional GUI, for engineers and process planners in industry to use and evaluate.

Future improvements on the algorithm include modifying the structure of the code to reduce and/or eliminate the effects of numerical errors, in order to remove the use of robustness countermeasures that compromises the optimality of the final solution. In order to further reduce cycle time, incorporation of SQP with LP+PWin is currently underway. Future meetings are being scheduled with engineers from the CNC company, Fanuc, to pinpoint the root-cause of profile distortions when pre-generated trajectories are fed to the CNC in inverse-time mode.

In 2D and 3D machining applications, the spline length can be easily described using the Pythagorean Theorem. In 5 axis machining, a tilt (A or B) and rotary (C) axis are added to define the orientation. Hence, toolpath geometry and derivatives cannot be simply defined by 3 Cartesian coordinates. This presents a problem when considering the path parameter in the constraint equations. Therefore, further research is required for the applicability of the LP+Pwin algorithm to 5-axis feedrate optimization.

References

[1] Zhang Q., Li S., Guo J.X., Gao X.S., 2015, "Time-optimal Path Tracking for Robots under Dynamics Constraints based on Convex Optimization", Robotica, Available on CJO 2015 doi:10.1017/S0263574715000247

[2] Fan, W., Gao, X., Lee, C. et al., 2013, "Time-optimal Interpolation for Five-axis CNC Machining along Parametric Tool Path based on Linear Programming", International Journal of Advanced Manufacturing Technology. 69(5), pp. 1373-1388

[3] Guo J.X., Zhang K., Zhang Q., Gao X.S., 2013, "Efficient time-optimal feedrate planning under dynamic constraints for a high-order CNC servo system". Computer-Aided Design, 45, pp.1538-1546.

[4] Okwudire C., Ramani K., Duan M., Hoshi T., 2016, "A trajectory optimization method for improved tracking of motion commands using CNC machines that experience unwanted vibration". CIRP Annals, 65(1), pp.373-376.

[5] Erkorkmaz K., 2012, "Real-Time Feed Optimization for Spline Toolpaths Employing Uninterrupted Acceleration Profiling", Proc. 27th ASPE Annual Mtg., San Diego.

[6] Lartigue, C., Tournier C., Ritou M., Dumur D., 2004, "High-Performance NC for HSM by means of Polynomial Trajectories", CIRP Annals – Manufacturing Technology. 53(1), pp. 317- 320.

[7] Wang F.-C., Yang D.C.H., 1993, "Nearly Arc-Length Parameterized Quintic-Spline Interpolation for Precision Machining", Computer Aided Design. 25(5), pp 281-288.

[8] Erkorkmaz K., 2015, "Efficient Fitting of the Feed Correction Polynomial for Real-Time Spline Interpolation", Journal of Manufacturing Science and Engineering. 137, pp. 044501(1)-044501(8)

[9] Sencer B., Ishizaki K., Shamoto E., 2015, "A Curvature Optimal Sharp Corner Smoothing Algorithm for High-Speed Feed Motion Generation of NC Systems Along Linear Tool Paths", The International Journal of Advanced Manufacturing Technology. 76(9), pp.1977-1992.

[10] Choi Y.H., Hong J.H., Jang S.H., 2005, "A Study on the Feed Rate Optimization of a Ball Screw Feed Drive System for Minimum Vibrations". ASME, 2, pp. 665-670.

[11] Barre P.J., Bearee R., Borne P., Dumetz E., 2005, "Influence of a Jerk Controlled Movement Law on the Vibratory Behavior of High-Dynamic Systems", Journal of Intelligent and Robotic Systems. 42(3), pp.275-293

[12] Brecher C., Lange S., Merz M., Niehaus F., Wenzel C., Winterschladen M., Weck M., 2006, "NURBS Based Ultra-Precision Free-Form Machining". CIRP Annals, 55(1), pp.547–550.

[13] Beudaert X., Lavernhe S., Tournier C., 2013, "5-axis local corner rounding of linear tool path discontinuities". International Journal of Machine Tools & Manufacture, 73, pp.9-16.

[14] Tulsyan, S., Altintas, Y., 2015, "Local toolpath smoothing for five-axis machine tools". International Journal of Machine Tools & Manufacture, 96, pp.15-26.

[15] Tsai M.C., Cheng C.W., Cheng M.Y., 2003, "A real-time NURBS surface interpolator for precision three-axis CNC machining". International Journal of Machine Tools & Manufacture, 43, pp.1217-1227

[16] Heng M., Erkorkmaz K., 2010, "Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability". International Journal of Machine Tools & Manufacture, 50, pp.281-293.

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

[18] Liu H., Liu, Q., Sun, P., Liu, Q., Yuan, S., 2017, "A Polynomial equation-based interpolation method of NURBS tool path with minimal feed fluctuation for high-quality machining". The International Journal of Advanced Manufacturing Technology, 90(9–12), pp. 2751–2759.

[19] Wang F.C., Wright P.K., Barsky B.A., Yang D.C.H., 1999, "Approximately Arcc-Length Parameterized C^3 Quintic Interpolatory Splines". ASME, 121, pp.430-439

[20] De Stantiago-Perez J.J., Osornio-Rios R.A., Romero-Troncoso R.J., Morales-Velazquez L., 2013, "FPGA-based hardware CNC interpolator of Bezier, splines, B-splines and NURBS curves for industrial applications". Computers & Industrial Engineering, 66(4), pp.925-932.

[21] Tajima S., Sencer B., 2016, "Kinematic corner smoothing for high speed machine tools". International Journal of Machine Tools & Manufacture. 108, pp.27-43.

[22] Sencer B., Ishizaki K., Shamoto E., 2015, "High speed cornering strategy with confined contour error and vibration suppression for CNC machine tools". CIRP Annals, 64(1), pp.369-372.

[23] Pateloup V., Duc E., Ray P., 2004, "Corner optimization for pocket machining". International Journal of Machine Tools and Manufacture, 44, pp.1343-1353.

[24] Yuen A., Zhang K., Altintas Y., 2013, "Smooth trajectory generation for five-axis machine tools". International Journal of Machine Tools and Manufacture, 71, pp.11-19.

[25] Chu C.N., Kim S.Y., Lee J.M., 1997, "Feed-rate Optimization of Ball End Milling Considering Local Shape Features". Annals of the ClRP Vol. 46, pp. 433-436.

[26] Ferry W.B., Altintas Y., 2008, "Virtual Five-Axis Flank Milling of Jet Engine Impellers – Part II: Feed Rate Optimization of Five-Axis Flank Milling". ASME, 130, pp.7-13.

[27] Ridwan F., Xu X., Churn F., Ho L., 2012, "Adaptive execution of an NC program with feed rate optimization". The International Journal of Advanced Manufacturing Technology, 63(9-12), pp.1117-1130.

[28] Xu K., Tang K., 2014, "Five-axis tool path and feed rate optimization based on the cutting force–area quotient potential field". The International Journal of Advanced Manufacturing Technology, 75(9-12), pp.1661-1679.

[29] Feng H.Y., Su N., 2000, "Integrated tool path and feed rate optimization for the finishing machining of 3D plane surfaces". International Journal of Machine Tools & Manufacture, 40, pp.1557-1572.

[30] Bobrow J.E., Dubowsky S., Gibson J.S., 1985, "Time-optimal Control of Robotic Manipulators Along Specified Paths", The International Journal of Robotics Research. 4(3), pp. 3-17.

[31] Pritschow G., 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, 1997.

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

[33] Erkorkmaz K., Heng M., 2008, "A Heuristic Feedrate Optimization Strategy for NURBS Toolpaths", CIRP Annals – Manufacturing Technology. 57(1), pp. 407- 410.

[34] Sun Y., Zhao Y., Bao Y., Guo D., 2015, "A smooth curve evolution approach to the feedrate planning on five-axis toolpath with geometric and kinematic constraints". International Journal of Machine Tools & Manufacture, 97, pp.86-97.

[35] Beudaert X., Lavernhe S., Tournier C., 2012, "Feedrate Interpolation with Axis Jerk Constraints on 5-Axis NURBS and G1 toolpath", International Journal of Machine Tools & Manufacture. 57, pp.73-82

[36] Erkorkmaz, K., Layegh, S.E., Lazoglu, I., Erdim, H., 2013, "Feedrate Optimization for Freeform Milling Considering Constraints from the Feed Drive System and Process Mechanics", CIRP Annals – Manufacturing Technology. 62(1), pp. 395–398.

[37] Sencer B., Altintas Y., Croft E., 2008, "Feed Optimization for Five-axis CNC Machine Tools with Drive Constraints", International Journal of Machine Tools & Manufacture. 48(7-8), pp. 733 – 745.

[38] 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 & Manufacture. 41(9), pp. 1323-1345.

[39] Liu M., Huang Y., Yin L., et al. 2014, "Development and implementation of a NURBS interpolator with smooth feedrate scheduling for CNC machine tools". International Journal of Machine Tools & Manufacture, 87, pp.1-15.

[40] Sun Y., Zhao Y., Bao Y., Guo D., 2014, "A novel adaptive-feedrate interpolation method for NURBS tool path with drive constraints". International Journal of Machine Tools & Manufacture, 77, pp.74-81.

[41] Altintas Y., Erkorkmaz K., 2003, "Feedrate Optimization for Spline Interpolation In High Speed Machine Tools". CIRP Annals, 52(1), pp. 297-302.

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

[43] Dong, J. and Stori, J.A., 2006. "A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization". Journal of dynamic systems, measurement, and control, 128(2), pp.379-390.

[44] Piegl L., Tiller W., 1997, "The NURBS Book". ISBN 978-3-642-59223-2.

[45] de Boor C., 1973, "Package for Calculating with B-Splines. SIAM Journal on Numerical Analysis". Los Alamos Scientific Lab., 4740859.

[46] Verscheure, D., Demeulenaere, B., Swevers, J., De Schutter, J. and Diehl, M., 2009. "Time-optimal path tracking for robots: A convex optimization approach". IEEE Transactions on Automatic Control, 54(10), pp.2318-2327.

[47] Kirk D.E.,1970, "Optimal Control Theory – An Introduction", Prentice-Hall NJ.

[48] Turbine Engine Solutions, "TURBINE ENGINE OVERHAUL", http://www.turbineenginesolutions.com/, accessed August 2017.

[49] Power Engineering, "Siemens Successfully Tests 3D-Printed Gas Turbine Blades" Feb 07, 2017, accessed August 2017. http://www.power-eng.com/articles/2017/02/siemens-successfully-tests-3d-printed-gas-turbine-blades.html