

**A REAL TIME AND TIME OPTIMAL TRAJECTORY GENERATOR FOR
CARTESIAN MACHINE TOOLS**

by

Sepehr Zarif Mansour

B.Sc., University of Tehran, 2013

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE COLLEGE OF GRADUATE STUDIES
(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Okanagan)

June 2015

© Sepehr Zarif Mansour, 2015

Abstract

This work focuses on control and trajectory optimization strategies for high speed contouring of machine tools. In the first part, control strategies are studied. Studied strategies are divided into three major categories. Axial error controllers, contour error controllers, and feed forward controllers. The control strategies are benchmarked on a biaxial XY table. The least contouring error was recorded for the control structure consisting of Cartesian Servo Control (CSC) with a Proportional Derivative Integral (PID) regulator + Torque Feed Forward (TFF). As for the trajectory optimization, a new real time algorithm to select time-optimal feedrates has been proposed. The algorithm is independent of the spline representation of the nominal path, and incorporates both velocity and acceleration constraints. The fact that the proposed algorithm is independent from the spline representation (since it simply adjusts sampling times for position increments), makes it more flexible than conventional algorithms in the literature that require particular spline representations of the reference trajectory. Also the proposed algorithm is computationally efficient because: 1) an analytical solution to the optimization is applied at every time step rather than a global numerical optimization procedure, and 2) other analytical solutions in the literature require a forward and a backward pass over the complete trajectory. The new algorithm only backtracks over a short window before decelerations. Finally, this study also introduces two new reference path generation techniques that use part tolerance values to reduce machining time.

Preface

A part of Chapter 5 is submitted to the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2015). I was responsible for the analytical derivation and experimental validation of all algorithms presented in this thesis, including the work submitted to the IDETC/CIE.

Table of Contents

Abstract.....	ii
Preface.....	iii
Table of Contents	iv
List of Tables	vii
List of Figures.....	viii
List of Symbols	xi
Acknowledgements	xiii
Dedication	xiv
Chapter 1: Introduction and Thesis Organization	1
1.1 Motivation.....	1
1.2 Objective	2
1.3 Thesis Outline	2
Chapter 2: Literature Review.....	4
2.1 CNC Machine Tools	4
2.2 Trajectory Generation	7
2.3 Control Strategies.....	9
Chapter 3: Experimental Comparison of Contouring Control Architectures.....	12
3.1 Model of the Experimental Biaxial Table	12
3.2 Cartesian Servo Control (CSC).....	13
3.1 Feed Forward Control Strategies	19
3.1.1 Torque Feed Forward (TFF) Controllers.....	19
3.1.2 The Zero Phase Error Tracking (ZPET) Controller.....	21
3.2 Cross Coupled Controller (CCC)	21
3.3 Tangential-Contouring (TC) Controller	23
3.4 Experimental Comparison of Control Strategies	25
3.4.1 Experimental Setup.....	25
3.4.2 Circular Interpolation.....	26

3.4.3	Corner Tracking.....	31
3.5	Discussions	34
Chapter 4: A New Real Time Feedrate Optimization Algorithm for 2-D Cartesian		
Machine Tools		35
4.1	Representing Variable Sampling Times in Position Control Systems with Constant Loop Closing Frequencies	35
4.2	Expressing Velocity and Acceleration for Sampled Data Systems with Uneven Sample Time	37
4.3	Sample Time Selection for a Single Axis	39
4.3.1	(1x) Limit Velocity in X	41
4.3.2	(2x) Limit Acceleration in X	42
4.3.2.1	(2xa) Limit acceleration:	42
4.3.2.2	(2xb) Limit Deceleration:	44
4.3.2.3	(2xc) Axis Reversal in X:	45
4.3.2.4	(3x) Backtrack in X	46
4.4	Algorithm for Sample Time Selection for Two Axes	48
4.4.1	(1xy) Limit Velocity in X and Y	51
4.4.2	(2x) Limit Acceleration in X, or (2y) Limit Acceleration in Y	51
4.4.2.1	(2xy) Limit Acceleration in X and Y Axes	52
4.4.2.2	(5) Backtracking in X and Y	55
4.5	Experimental Setup	56
4.6	Experimental Results	57
4.7	Discussions	63
Chapter 5: Reference Trajectories with Tolerance Bounds		64
5.1	Tolerance Bounds at Corner Points	64
5.2	Case Studies	65
5.2.1	Circles	65
5.2.2	Single Axis Reversal:	67
5.2.3	General Corner	68
5.3	Fitting an Arc in the Corner	70
5.4	Path Modification Using Average Filter	71
5.5	Experimental Setup	73
5.6	Experimental Results	74

5.7	Discussions	76
Chapter 6: Conclusions		77
6.1	Experimental Comparison of Contouring Control Architectures	77
6.2	A New Real Time Feedrate Optimization Algorithm for 2-D Cartesian Machine Tools ...	78
6.3	Reference Trajectories with Tolerance Bounds	78
6.4	Future Works	79
References		81

List of Tables

Table 3. 1	CSC comparison	15
Table 3. 2	Experimental setup	26
Table 3. 3	Circular trajectory parameters	27
Table 4. 1	Experimental setup	57
Table 5. 1	Experimental setup	73
Table 5. 2	Experimental results using nominal reference trajectory and the feedrate override method in Chapter 4.	75
Table 5. 3	Experimental results of two proposed algorithms.	76

List of Figures

Figure 2. 1	Vertical milling machine	5
Figure 2. 2	Center lathe.....	5
Figure 2. 3	Block diagram of a CNC machine's servo axis.....	6
Figure 2. 4	Contouring error and axial errors [43].....	7
Figure 3. 1	Block diagram of the high speed XY table	13
Figure 3. 2	CSC strategies a) State feedback (SF), b) Proportional Derivative (PD), c) Proportional Integral Derivative (PID)	15
Figure 3. 3	Group delay for $\omega_c=30\pi$ rad/sec.....	17
Figure 3. 4	Control loop magnitude response for $\omega_c=30\pi$ rad/sec.	18
Figure 3. 5	Disturbance magnitude response for $\omega_c=30\pi$ rad/sec	19
Figure 3. 6	Torque feed forward control block diagram.....	20
Figure 3. 7	ZPET block diagram.....	21
Figure 3. 8	Axial, contour, and tangential errors for a linear contour	22
Figure 3. 9	CCC block diagram	23
Figure 3. 10	TC block diagram [43]	24
Figure 3. 11	Biaxial table used for the experiments	25
Figure 3. 12	Axes velocities.....	27
Figure 3. 13	Path (feed) velocity.....	28
Figure 3. 14	Contouring errors of CSC, LCCC and TC controllers	29
Figure 3. 15	Circular contour errors for ZPET controllers and torque feed forward controllers	30
Figure 3. 16	Corner tracking trajectory.....	32
Figure 3. 17	a: Axis velocities. b: Tangential velocity	32
Figure 3. 18	Contouring errors for corner tracking.....	33
Figure 4. 1	a) X axis moves with programed velocity. b) Feedrate modulation by inserting zero motion increments to reduce X axis's velocity to half of the programed velocity. c) Combining every fifth sample with its predecessor and 5/4 increase in the programed velocity.	36
Figure 4. 2	Parabolic position interpolation.....	38

Figure 4. 3	Flowchart of the single axis algorithm	40
Figure 4. 4	Velocity constraint.....	41
Figure 4. 5	Acceleration constraint.	43
Figure 4. 6	Deceleration constraint.	44
Figure 4. 7	Axis reversal constraint.	45
Figure 4. 8	Decelerating from maximum velocity to zero.....	47
Figure 4. 9	Flowchart of the two axes algorithm	50
Figure 4.10	Violated acceleration in X and constrained acceleration in Y (black lines). Simultaneously limiting acceleration in X and Y axes by increasing the sample times (red lines)	52
Figure 4. 11	Control system.....	57
Figure 4. 12	A teardrop curve is used as the original reference path.....	58
Figure 4. 13	Original reference path, velocity and acceleration profiles of the teardrop.	59
Figure 4. 14	Reference and actual velocity.....	60
Figure 4. 15	Reference and actual acceleration.	60
Figure 4. 16	Reference and actual acceleration at the corner of the trajectory.....	61
Figure 4. 17	Contour error	62
Figure 5. 1	Tolerance boundary lines and tolerance circle (shown in red) around the tool path (shown in black) [70].....	65
Figure 5. 2	Traversing along a circle a) along the inner tolerance band b) along the outer tolerance band.....	66
Figure 5. 3	Single axis reversal a) along programmed path b) along outside tolerance band.....	67
Figure 5. 4	Corner tracking a) along the inside tolerance band, b) between outside and inside tolerance band.....	69
Figure 5. 5	Travel time for a general corner with inside angle θ	70
Figure 5. 6	Fitting an arc in the corner.....	71
Figure 5. 7	Using average filter to modify a path	72
Figure 5. 8	Nominal path, modulated reference path, and actual path for the first strategy (fitting an arc in the corner)	74

Figure 5. 9	The nominal path, the modified reference path, and the actual path for the second strategy (using running average filter).....	75
-------------	--	----

List of Symbols

$a_{max,normal}$	Maximum normal acceleration on a circle
b	Viscous damping coefficient
ε_c	Contour error
ε_t	Tangential error
r_{circle}	Radius of circular reference path
$j_{max,x}$	Maximum number of backtracking steps
$p_x p_y$	Ball screw pitch
$t_{modulate}$	Minimum time required to decelerate from maximum to zero speed
v_{circle}	Circular motion velocity
ω_c	Controller natural frequency
ω_{circle}	Angular frequency of reference trajectory
C_F	Control loop regulator
C_L	Feedback regulator
CCx	Cross coupling gain for the X axis
CCy	Cross coupling gain for the Y axis
$Cx(s), Cy(s)$	Actual positions in x and y directions
$D(s)$	Transfer function between disturbance torque and actual position
E_x, E_y	Axial errors in x and y directions
$I_{max,x}, I_{max,y}$	Maximum Motor currents in x and y directions
J_x, J_y	Axis Inertias in x and y directions
K_d	Control loop derivative gain
K_i	Control loop integral gain
K_{tx}, K_{ty}	Motor constants in x and y directions
K_p	Control loop proportional gain

N_{max}	Maximum length of a running average filter
$P(u)$	Parameterized curve
R_x, R_y	Motor resistances in x and y directions
$Rx(s), Ry(s)$	Input reference positions in x and y directions
SF	Scale factor
Tol	Allowable machining tolerance value
$T(s)$	Overall position loop transfer function
T_g	Group delay
$T_d(s)$	Disturbance torque
T_m	Motor torque
S	The distance required to decelerate from Maximum to zero speed
$U_{max,x}, U_{max,y}$	Maximum motor voltages in x and y directions
$\dot{X}, \dot{Y}$	Axis velocities in x and y directions
$\dot{X}_{pro}, \dot{Y}_{pro}$	Programed velocities in x and y directions
$\ddot{X}, \ddot{Y}$	Axis accelerations in x and y directions
Δt_a	Sample time required to meet acceleration constraints
Δt_b	Sample time required for backtracking
Δt_R	Sample time required for axis reversal
Δt_{nom}	Nominal sample time
Δt_v	Sample time required for velocity constraints
$\Delta X, \Delta Y$	Axis motion increments in x and y directions

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Rudolf Seethaler. It was a great honor for me to work with a professional and friendly supervisor. I am thankful for your patience, valuable advice, and your support throughout my Master's studies. It was not possible for me to complete and write my thesis without your guidance and encouragement.

I would also like to thank Dr. Ian Yellowley for supporting my Master's studies.

Last but not least, I would like to offer my special thanks to my beloved family for their endless support and love.

Dedication

*To my mother, who taught me to be diligent,
To my father, who taught me to be tolerant,
To my sister, who is my spiritual support.*

Chapter 1: Introduction and Thesis Organization

Automation is a process in which a machine follows a specific order of steps in order to carry out a desired task with small or no human interaction [1]. This concept has made considerable changes to industrial manufacturing processes and has been implemented in different stages: 1- manufacturing operations (milling, drilling, shaping, etc.), 2- moving components, 3- assembling components, 4- packaging the products [1]. Within the manufacturing operations a hierarchical process plan is followed that prescribes the type of machine tools to be used, the tool paths to be followed and finally the selection of machining parameters such as feedrate and spindle speed. The topic of this thesis deals with the optimization of the algorithms employed by the machine tool that translates the low level process plan (path and feedrate) to machine tool motions.

1.1 Motivation

Controlling a machine tool to follow the process plan and produce high quality parts is a challenging topic. Today, Computer Numerically Controlled (CNC) machine tools are commonly used in manufacturing environments. They are usually programmed with a programming language referred to as G-code, which was defined in the 1960s and in its simplest form, contains geometrical information (tool path) and machining parameters (eg feedrate and spindle speed). The process planning system usually has very little knowledge about the physical capabilities of the machine tools, and thus machining parameters are commonly selected by the operator from experience. This leads to inefficiencies since operators generally want to err on the conservative side in order to operate the machine safely and produce little waste in parts and tools. Thus, there is considerable potential to improve machining parameters. In fact, there is a whole literature on optimizing process plans. These optimization procedures however are usually very complex and need to be performed in a preprocessing step that is dependent on precise material data and machine performance data that often is not available.

Thus, there is a clear need for computationally efficient real-time control strategies that are hosted in the machine tool itself. These control strategies should be able to increase accuracy

while reducing machining time. In the case of ill-defined material properties and raw part geometries, the algorithms need to be able to adapt machining parameters to ensure safe, efficient and accurate parts.

1.2 Objective

This thesis focuses on optimizing the low level process plan (feedrate and toolpath) and its interaction with the position control strategy implemented in the machine tool. To this end, the performance of a wide variety of different position control strategies are analyzed analytically and compared experimentally in order to synthesize the most efficient and practical low level control strategy. The result of this comparison provides the basis necessary for developing a new real-time feedrate optimization algorithm that is validated experimentally using linear, circular and freeform motion segments. Contrary to existing feedrate optimisation algorithms, the new algorithm needs to be analytical to ensure computational efficiency for real-time implementation. In addition, the algorithm needs to be able to work with a finite lookahead window of the path, since the memory for motion increments on machine tools is limited. The final and third objective of this thesis is to provide an algorithm that modifies the programmed path in order to take advantage of machining tolerances when optimizing for minimum machining time.

1.3 Thesis Outline

The main goal of this study is providing machine tool manufacturers with contouring strategies which are easy to implement on their machines. A critical review of position control strategies and -feedrate selection strategies is provided in Chapter 2. In Chapter 3 the contouring performance of different controller architectures which are provided with physically feasible reference trajectories, are compared experimentally. More specifically, the relative performance of conventional axis servo drives, contour error control schemes, and feed forward control schemes are investigated. Chapter 4 proposes a new real time feedrate optimization algorithm that will provide time-optimal and feasible Cartesian trajectories, constrained by velocity and acceleration limits. The algorithm is intended to

reside inside the CNC controller. Chapter 5 introduces new trajectory generation algorithms which use predefined tolerances to smooth trajectories and reduce machining time. Chapter 6 provides conclusions and suggestions for future work.

Chapter 2: Literature Review

Modern Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) CAD/CAM systems create the process plan for producing a part by selecting machining operations, tools, tool paths, feedrates, and spindle speed. The process plan is the result of a complex multi-level optimization algorithm based on incomplete machine, material, and geometry data. This thesis aims to provide novel real-time algorithms that update and optimize the lowest levels of the process plan (feedrate and toolpath) during the actual machining operation. In this chapter, a very brief overview of common machine tools and their sources of machining errors is provided followed by a review of path planning, feedrate selection, and motion control strategies.

2.1 CNC Machine Tools

The process plan generated by modern CAD/CAM systems has many levels. The top level is responsible for selecting machining operations. There are various types of CNC machines for machining operations. Two representatives are milling machines and lathes shown in Figure 2.1 and 2.2 respectively. CNC lathes, are used for machining cylindrical parts. The parts are mounted in a rotating spindle, and the tools are fed into the parts at a feedrate that is measured on mm per spindle revolution. CNC milling machines can be used to create almost any geometry. Their tools are mounted on the rotating spindle and the work piece is fed into the rotating tool.



Figure 2. 1 Vertical milling machine



Figure 2. 2 Center lathe

The second level of the process plan contains selection of tools (size and type) and tool paths. The toolpaths define the depths and width of cuts taken by the machine tool in order to create the final desired geometry. The third level of the process plan contains the selection of spindle rotating speed and feedrate.

In motion control research, it is common to distinguish between a path and a trajectory. The path contains geometry only (waypoints of the tool with respect to the part). The trajectory includes waypoints and timestamps. The toolpath describes the part geometry and the trajectory describes the dynamic behavior of the machine tool.

The process planning system delivers a nominal trajectory to the machine tool consisting of a nominal path that can be line segments, arcs, or splines and a nominal feedrate. The machine tool then employs an interpolator that creates a reference trajectory consisting of motion increments that are equally spaced in time and ready to be fed to the position control loop for each axis of motion. Figure 2.3 shows a block diagram of a CNC machine's servo axis.

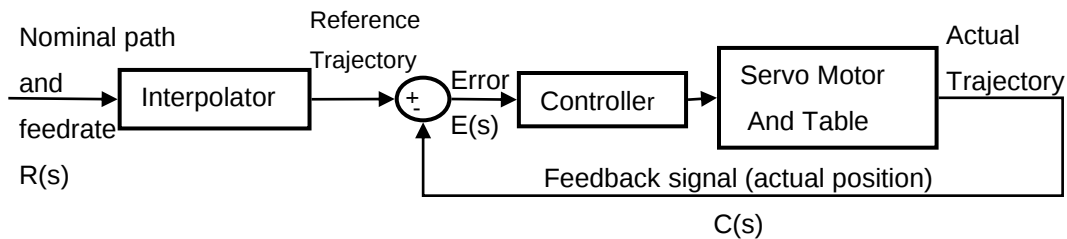


Figure 2. 3 Block diagram of a CNC machine's servo axis

The objective of a CNC machine tool is to follow the nominal path provided by the process planning system with as little contouring error as possible. Due to limited system dynamics and nonlinearities, the control system cannot always exactly follow the reference path. The resulting errors in position can be expressed in two different coordinate systems as shown in Figure 2.4. Contour error (ϵ_c), measures the minimum distance between the actual point and reference path. Axial errors (E_x) and (E_y), represent distances between the actual point and reference position in the direction of the physical drive axes X and Y respectively.

The axial error shown in Figure 2.4 assumed that the machine tool uses a Cartesian drive system. This is true for many machine tools.

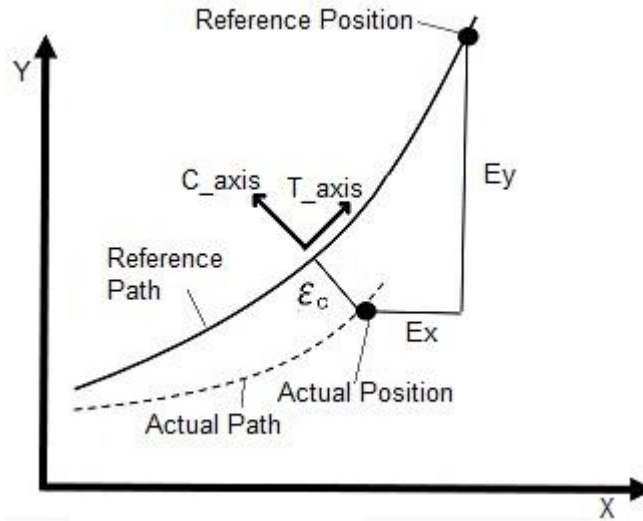


Figure 2.4 Contouring error and axial errors [43]

Contour error can be caused by improper design of the machine tool (alignment, temperature, stiffness), or deficiencies of the machine tool software, which performs two basic tasks. First, the interpolator creates feasible reference trajectories that do not violate velocity and acceleration limitations of the axis servo drives. Second the control system needs to track the feasible reference trajectories as closely as possible. The next two sections review the available literature that has been created for these tasks.

2.2 Trajectory Generation

Motion increments of the reference trajectory of each axis are generated by geometric interpolators: e.g. linear/circular interpolators, or parametric interpolators for freeform surfaces. To reduce acceleration levels at the start and end of the motion segments, Acceleration/Deceleration algorithms are commonly incorporated into the interpolators for lines, circles and splines in commercial machine tools [3]. However, these commercial implementations are usually very conservative and they only limit accelerations at the beginning and the end of motion segments.

The machining literature shows a wealth of different approaches to further improve trajectories [3-16]. Farouki et al. [11] proposed a closed loop solution for the feedrate optimization problem using degree- n Bezier trajectories. Their algorithm provides feedrates

for the two cases: constant, and speed dependent acceleration limited trajectory. A limitation of the proposed algorithm is that it assumes the same acceleration limits in the X and Y directions. Jingchun Feng et al. [12] proposed a real time NURBS interpolator with axis acceleration limits for biaxial tables, which confines axis acceleration, and chord error, and provides an analytical solution for the time optimal trajectory problem. Altintas and Erkokmaz [13] developed a quintic spline trajectory generation algorithm using jerk, acceleration, and velocity limits. There are also several other studies which take jerk constraints into account to develop smooth acceleration trajectories [8,14,15]. However, adding the jerk to the constraints increases computation time, since numerical search algorithms have to be used in place of analytical solutions for finding minimum time feedrates. The algorithms shown for parametric interpolators above [3-15] are complex and require a particular spline representation (Polynomials, NURBS, Bezier, and etc.) of the desired geometry. Thus, they can be difficult to implement, since they require that both the CNC machine and the CAD system use the same interpolation scheme.

Dong and Stori [16] presented a generalized time optimal feedrate generation algorithm which is independent of the spline representation. For a given arbitrary parameterized curve $P(u)$, $0 \leq u \leq 1$ (the original reference path), with velocity and acceleration constraints, the algorithm uses a two pass iterative search algorithm to find the global time optimal feedrate. The forward pass limits acceleration and the backward pass limits decelerations. A detailed proof of optimality is also provided for the proposed algorithm. One drawback of this algorithm is that it is difficult to implement in real time machining operations since it requires the complete trajectory at the beginning of the task.

All the interpolators discussed above follow the nominal reference path as closely as possible, without creating voltage or current violations. Thus, they have to come to a complete stop at critical points such as corners. Another class of parametric interpolators use predefined tolerance bounds around the original reference path, and instead of exactly following the nominal path, they create the reference path within the predefined tolerance bounds. This strategy relaxes the acceleration requirements of the reference path at sharp corners and makes the machine move faster. Bosetti and Bertolazzi [72] used variable tolerances at different portions of the path. However, they require a computationally expensive optimization algorithm to determine optimal feedrates and reference paths. Charlie

et al. [73], also used tolerance at the critical portions of the nominal reference path like sharp corners to create a smooth reference path. However, their algorithm is also computationally expensive.

In summary, commercial machine tools use conservative acceleration/deceleration algorithms for ensuring low contouring errors at the start and end of lines and circles. Some more advanced machine tools can also create splines. However, due to the substantial computational cost, advanced splines with limited velocity, acceleration and jerk are not implemented in commercial machine tools.

2.3 Control Strategies

The position control system in a machine tool aims to guide the actual position along the reference path with as little contouring error as possible. In its simple form, the control system first calculates position errors from the difference between reference and actual positions. Then it uses this error to decide on an appropriate control action to drive the servo motor with. One can either calculate control actions in the directions of the physical drive axes, or normal and tangential to the path. Due to simplicity and robustness, commercial machine tools use axial drive errors with State Feedback (SF) controllers, Proportional Derivative (PD) controllers, or Proportional Integral Derivative (PID) controllers.

The literature shows a great number of more advanced control strategies for compensating the axial error. Since most of them are computationally expensive and have not been proven to provide enough benefit, they are rarely used by commercial machine tools. However, they are important to understand what is technically feasible.

Altintas and Erkorkmaz [63], introduced an adaptive sliding mode controller which is robust against nonlinearities such as coulomb friction and disturbances. The major characteristic of this controller is designing a “sliding surface (S)” as a function of bandwidth of the system. A Lyapunov function is used to prove the stability of the controller under nonlinear disturbances. One drawback of this control scheme is that due to the switching nature of this control scheme, actuator chatter can be a concern.

Hao and Limei [68], used a real time Model Predictive Control (MPC) strategy on a biaxial table. MPC generates control commands by optimizing the predicted model behavior in an

iterative manner [69]. However, computational cost make this algorithm hard to implement on commercial industrial machines.

Jee and Koren [64] introduced an adaptive fuzzy logic controller. This algorithm replaces the conventional model based control strategies with rule based ones. The rationale for this approach is that unknown models for friction and nonlinear disturbances make model based controllers perform poorly, but, rule based controllers do not need complete model information. Jee and Koren use three parts to define their control strategy. During “fuzzification”, control inputs are converted to fuzzy variables. Then the “inference engine with a rule base” creates a fuzzy output based on a set of predefined rules. Finally, during “defuzzification”, the fuzzy output is converted to a non-fuzzy controller output. Again commercial success of this and other fuzzy based algorithms has been hampered by concerns of robustness and computational efficiency.

The axial controllers described so far in this section are all feedback controllers. In the robotics literature, tracking controllers are used to achieve smaller contour errors by adding a model based feedforward term to the control strategy. Surprisingly, for machine tool applications, only Zero Phase Error Tracking (ZPET) controllers [25-27], a particular variant of feed forward controllers have been studied widely. Traditional computed Torque Feed Forward (TFF) [34-37, 56, 57] strategies have mainly been studied in connection with robotic tracking. All feedforward tracking control strategies aim to force individual axis error to zero. This approach is particularly successful in reducing transient errors, where the bandwidth of the feedback control system is too low to capture the frequency content of the reference trajectory.

In order to reduce contour error without forcing the individual axis error to zero, Koren proposed Cross Coupled Control (CCC) [28]. The original algorithm only works well for straight line reference trajectories. By using time variant gains one can also account for arbitrary curvatures [29]. A central part in this algorithm is the real-time calculation of contour error. Many studies attempt to improve calculation of contour error for CCC [24, 30, and 40]: Shih et al. [30] use the Taylor series expansion. Jeremy R. Conway et al. tried to compute the exact value of the contouring error for general free form trajectories through a polynomial root tracking algorithm [40]. Other studies replace or augment the PID controller of CCC with more advanced control schemes. Ke-Han Su et al. and Ming-Yang Cheng

improved contouring error accuracy by adding a position error compensator to the CCC structure, which simultaneously reduces the axial and contouring errors [21]. Syh-Shiuh Yeh and Pau-Lo Hsu [33] proposed a robust CCC design, which provides a guaranteed control performance [32-34]. Feed forward controllers have also been employed to improve the CCC algorithm [34-37]. Yet another control strategy combines Iterative Learning Control with CCC [38]. Barton et al. [39] used this combination and introduced a control strategy for repetitive processes. The main feature of this algorithm is using the output of the system at each iteration to update the control command for the next iteration. The main advantage of the proposed algorithm is that it is able to learn from the previous iterations to improve its performance in the presence of unmodelled disturbances [39]. However, the algorithm is limited to the repetitive machining process.

Contrary to axial error controllers the so called task coordinate frame approach calculates control errors in the contour (normal) and tangential directions of the tool path [42-50]. There are many ways to define the contour and tangential errors. Yao's group calculates the contour error based on geometric characteristics of the reference trajectory called a global task coordinate frame [48], [49]. Yunjiang Lou et al. proposed a task polar coordinate frame that estimates the contour error using a second order approximation [50]. Recently the task coordinate frame approach has been paired with advanced robust and adaptive control schemes. Hu et al. [46] employed an adaptive robust controller combined with a task coordinate frame in order to improve the system's performance against friction and nonlinearities. They have specifically studied the performance of their control strategy against cogging force. The Adaptive Robust Control (ARC) framework that they used is based on work by Yao and Tomizuka [65-67]. The framework adapts to and is robust against unknown nonlinear disturbances. However, it is computationally expensive, which makes it difficult to implement on commercial machine tools.

In summary, axial error control, cross coupled control, and task coordinate frame control are the main control architectures used for machine tool. Within these architectures, conventional PID, advanced MPC, adaptive control, and feed forward strategies can be applied. Due to computational costs and robustness, industrial machine tools mainly rely on axial error control with a PID regulator.

Chapter 3: Experimental Comparison of Contouring Control Architectures

This chapter provides an experimental comparison of contouring control architectures. In particular, conventional axis servo control, cross coupled control, and task coordinate frame control are studied. The comparison presented in this chapter is based on SF, PD, and PID feedback controllers. This choice of controllers allows for a physical explanation of the relative performance of the overlying control architectures using frequency response arguments. In addition, feedforward tracking strategies are added in order to provide better transient performance. Before providing details of the experimental results, the model of the experimental biaxial table is presented and the different control systems are discussed in detail.

3.1 Model of the Experimental Biaxial Table

A block diagram of the high speed biaxial table is shown in Figure 3.1. The table has different inertias J_x , and J_y in the X and Y directions respectively. The inertias are driven through ball screws by identical brushed DC motors that provide torque, T_m , proportional to current, I :

$$T_m = K_t \times I \quad (3.1)$$

where, K_t is the motor torque constant. Motors are driven by current amplifiers, whose bandwidths are tuned to be more than an order of magnitude higher than the bandwidths of the position controllers. As a result, reference and actual currents appear identical to the position controllers, and the current amplifier will be omitted in the derivations during the remainder of this thesis. The overall plant transfer function is then described by:

$$G(s) = \frac{K_t}{Js^2 + bs} \quad (3.2a)$$

$$G(s) \sim \frac{K_t}{Js^2}. \quad (3.2b)$$

Even though viscous friction, b is a common phenomenon in heavy machine tools [71], it plays little role in high speed X-Y tables used for applications such as laser machining. For the XY-table used in this study, the friction term can be neglected.

The position controllers provide a desired current to the current amplifiers for each axis. Depending on the architecture of the position controller (shown as XY position controller in Figure 3.1), the reference current can be derived from axial errors, or contour errors.

The overall position loop transfer function will be denoted as $T(s)$ in this work:

$$T(s) = \frac{C(s)}{R(s)}. \quad (3.3)$$

Disturbances are accounted for through the disturbance torque, T_d . For later reference, the transfer function between T_d and resulting actual position is denoted as $D(s)$:

$$D(s) = \frac{C(s)}{T_d(s)}. \quad (3.4)$$

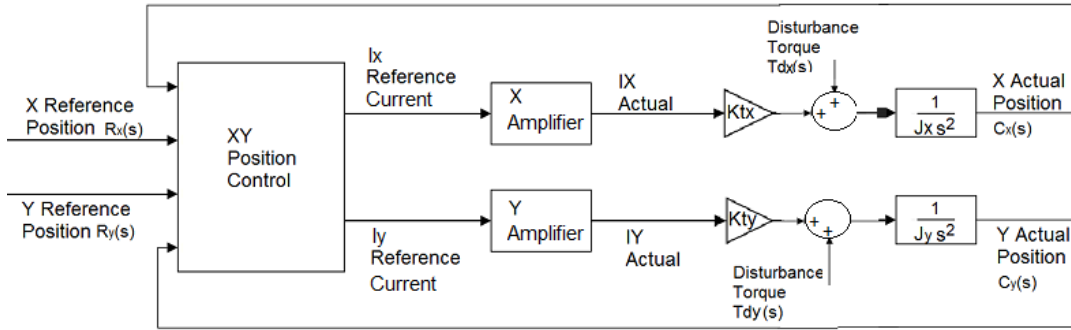
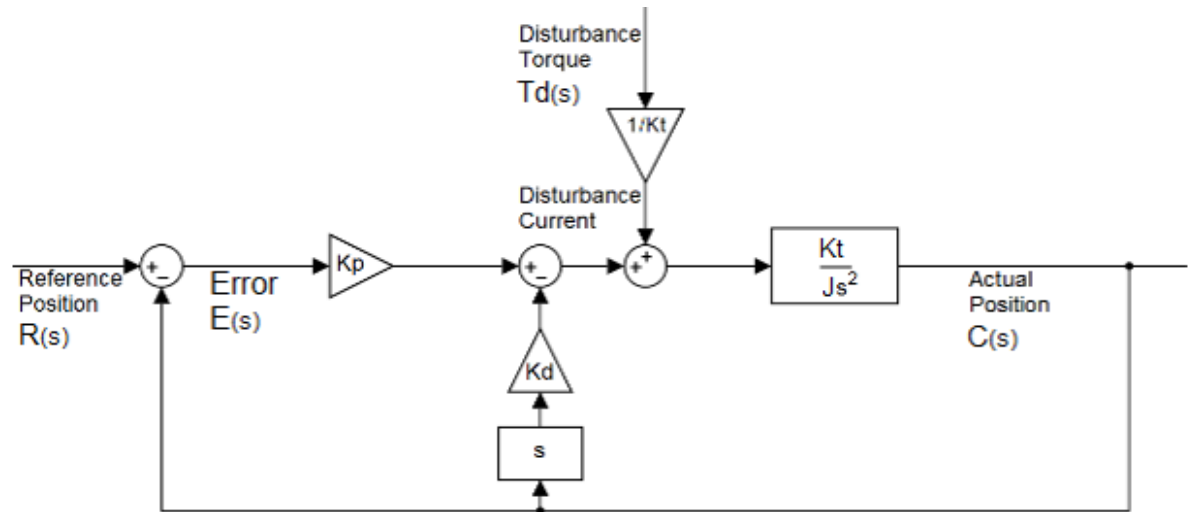


Figure 3. 1 Block diagram of the high speed XY table

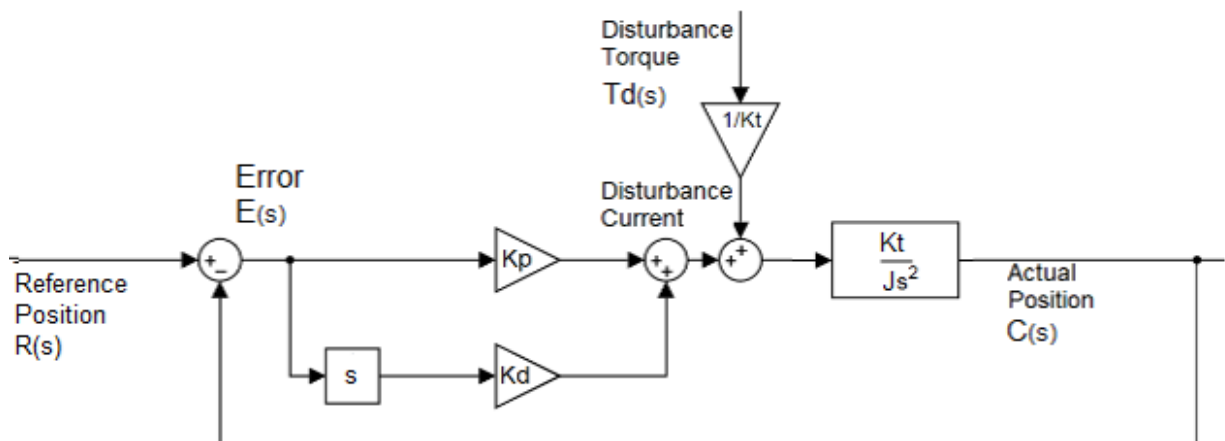
3.2 Cartesian Servo Control (CSC)

The most common control strategies for machine tools use independent servo controllers for each drive axis. This thesis focuses on the common case of Cartesian drive systems. Thus, axial error controllers are referred to as Cartesian Servo Controllers (CSC). Tracking performance and disturbance rejection is dependent on the particular control law employed.

The three most widely used commercial CSC strategies are State feedback (SF), Proportional Derivative (PD) and Proportional Integral Derivative (PID) control. Figure 3.2a-c shows block diagrams of these controllers.



a



b

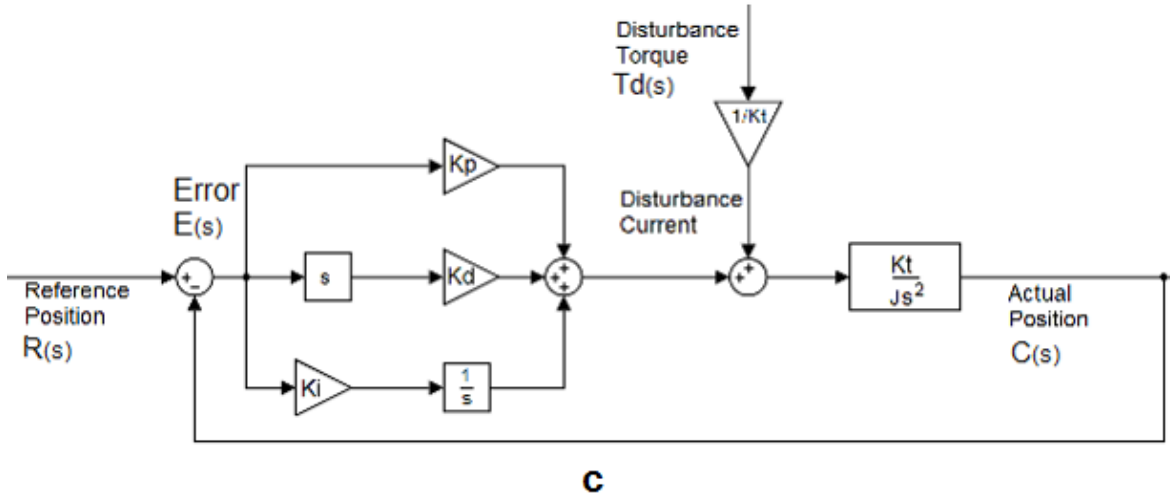


Figure 3. 2 CSC strategies a) State feedback (SF), b) Proportional Derivative (PD), c) Proportional Integral Derivative (PID)

The SF controller considered in this paper consists of two parts: a proportional gain acting on the position error, and a derivative gain acting on the measured velocity. Contrary to the SF controller, the derivative gain for a PD controller is not applied to measured velocity, but to the derivative of position error. In addition to the proportional and derivative components of SF and PD controllers, the PID controller employs an integrator.

To achieve a fair comparison between the three control strategies, the controller gains are selected to provide identical real poles at a frequency of $\omega_c = 40\pi \text{ rad/sec}$ as shown in Table 3.1. The SF and the PD control laws lead to a second order system, with identical denominator. The only difference between the two controllers is the additional zero for the PD case. The PID control law leads to a third order system, with two zeros.

Table 3. 1 CSC comparison

	$T(s)$	K_p		K_d	K_i	$\lim_{\omega \rightarrow 0} T_g$	$\lim_{\omega \rightarrow 0} D(j\omega)$
SF	$\frac{K_p K_t}{Js^2 + (K_d K_t + b)s + K_p K_t}$	$\frac{\omega_c^2 J}{K_t}$		$\frac{2\omega_c J}{K_t}$	0	$\frac{2}{\omega_c}$	$\frac{K_t}{\omega_c^2 J}$
PD	$\frac{K_d K_t s + K_p K_t}{Js^2 + (K_d K_t + b)s + K_p K_t}$	$\frac{\omega_c^2 J}{K_t}$		$\frac{2\omega_c J}{K_t}$	0	0	$\frac{K_t}{\omega_c^2 J}$

PID	$\frac{K_d K_t s^2 + K_p K_t s + K_i K_t}{Js^3 + (K_d K_t + b)s^2 + K_p K_t s + K_i K_t}$	$\frac{3\omega_c^2 J}{K_t}$		$\frac{3\omega_c J}{K_t}$	$\frac{\omega_c^3 J}{K_t}$	0	0
-----	---	-----------------------------	--	---------------------------	----------------------------	---	---

One common source of error in contouring control is axis mismatch. In terms of the control literature, this phenomenon can be explained using group delay. Ideally, the control loop should be linear in phase with identical group delay, T_g , for all axes:

$$T_g = \frac{1}{\omega} \tan^{-1} \left(\frac{\text{Imag}(T(j\omega))}{\text{Real}(T(j\omega))} \right). \quad (3.5)$$

Figure 3.3 shows that the group delay for trajectories with low frequency components is 16ms for the SF controller and 0 ms for the PD and the PID controllers. Table 3.1 lists the analytical expressions for this zero frequency group delay, which is responsible for the time-lag between reference and actual positions during linear motion segments. Axis mismatch is the phenomenon whereby the zero frequency group delay is not identical for all axes, which leads to contouring errors in linear motion segments. The SF controller can be tuned to minimize this effect, but both the PD and PID controllers automatically avoid axis mismatch during linear motion segments.

For circles, both motion axes experience the same frequency components, and a matched controller will provide identical time lags between reference and actual position for all axes. However for generalized curvilinear motion segments, time lag is a function of the frequency content of the desired trajectory. To minimize contour error one would need to ensure that the frequency components of the trajectories are well below the bandwidth of the position control loops, where the controllers have approximately linear phase.

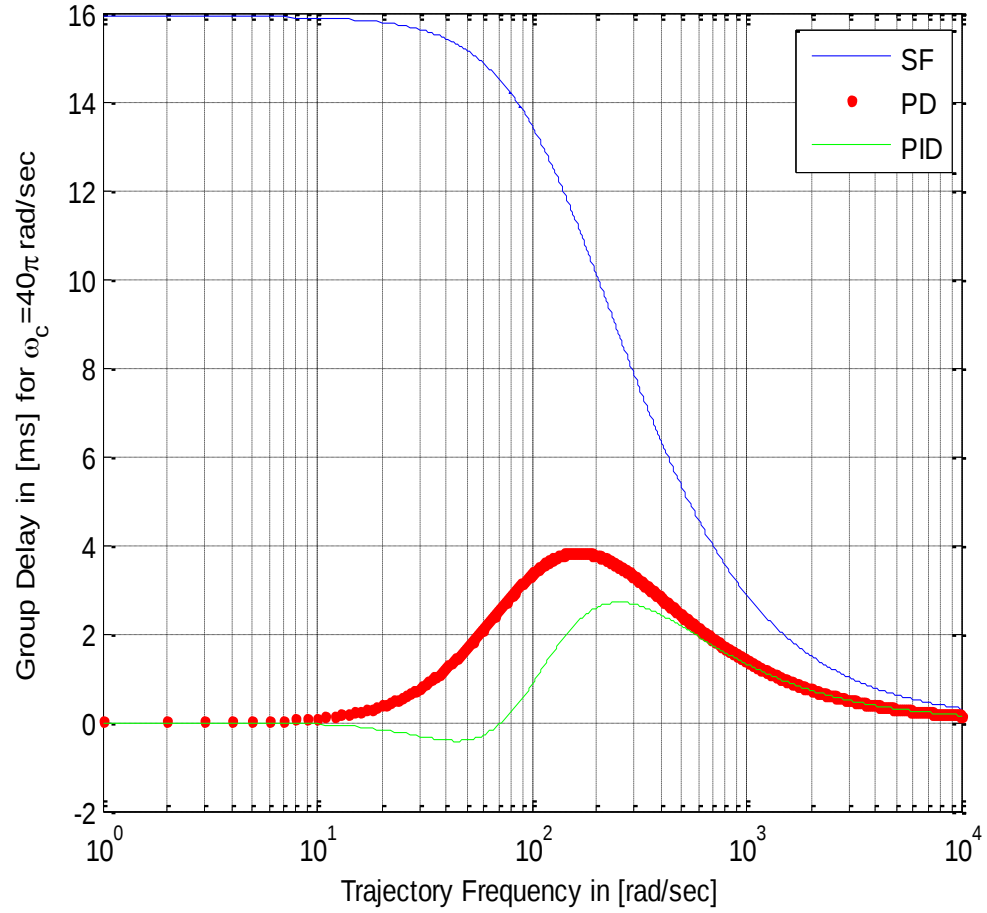


Figure 3. 3 Group delay for $\omega_c=30\pi$ rad/sec

Axis mismatch is related to the phase response of the position control loop. The magnitude response can also lead to contour errors, when the reference trajectory contains frequencies close or beyond the bandwidth of the control loop. Figure 3.4 indicates that the magnitude response for the three systems is close to unity for frequencies well below the bandwidth of the position loop. However, for frequencies close to ω_c , the SF controller's gain is less than unity and the gains for the PD and PID controllers are larger than unity. Clearly, during fast circular interpolation, the SF controller's circles will be too small and the circles of the PD and PID controllers will be too large.

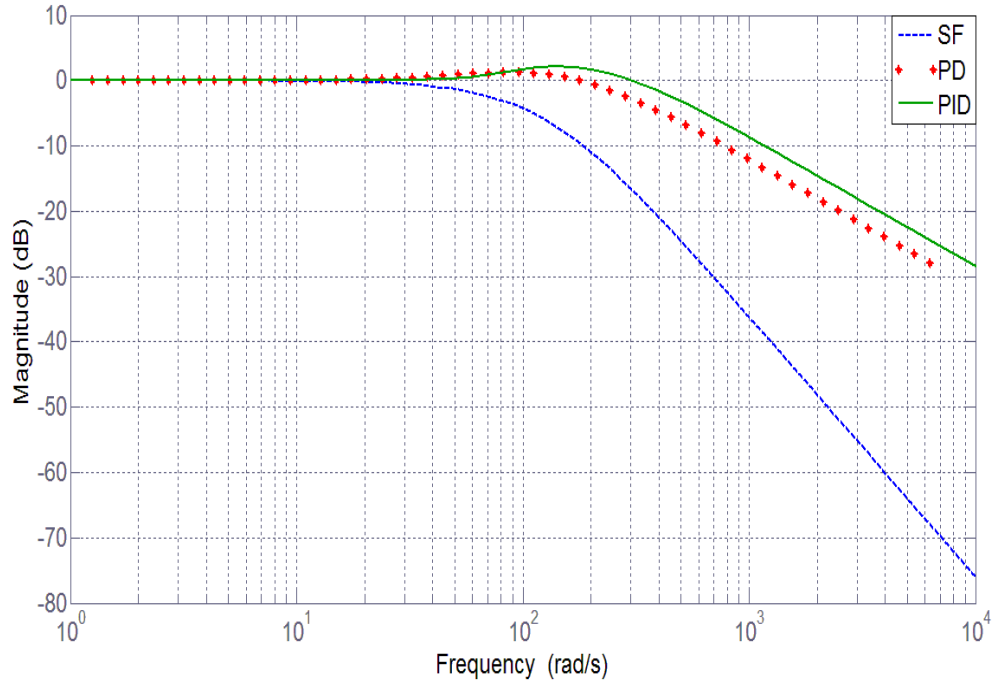


Figure 3. 4 Control loop magnitude response for $\omega_c=30\pi$ rad/sec

Finally, un-modelled disturbance torques can also lead to significant contouring error. This topic is investigated in depth by a robust control research community, which aims to guarantee asymptotic stability for different types of bounded disturbances. In this thesis, this problem is addressed by comparing the magnitude response of the disturbance transfer function, $D(s)$. In Figure 3.5 PID control demonstrates better disturbance rejection across all frequencies up to the bandwidth of the position loop. Beyond the position loop bandwidth, all three controllers (SF, PD, and PID) display almost identical disturbance rejection. Hence, compared to the SF and PD controllers, the PID controller is expected to have the lowest errors throughout the usable frequency spectrum of the controllers. Table 3.1 lists the steady state position response to a constant unity disturbance torque. Only the PID controller rejects constant disturbance torques. Thus, only a PID controller is able to reject the common phenomenon of sticking friction and provide zero error when the axes are not moving.

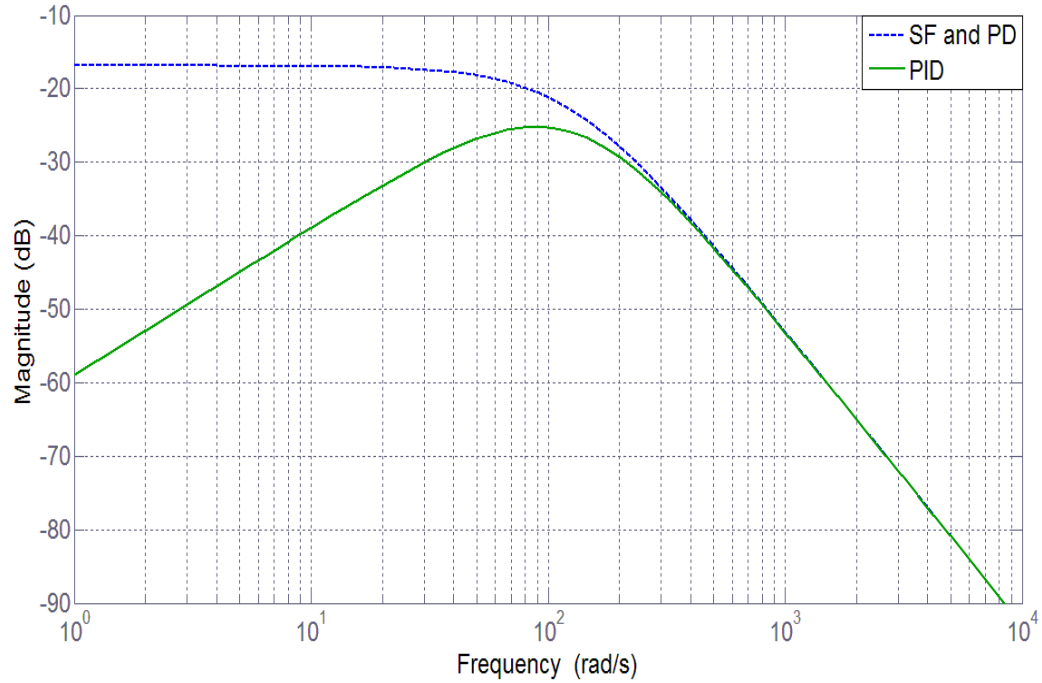


Figure 3. 5 Disturbance magnitude response for $\omega_c=30\pi$ rad/sec

3.1 Feed Forward Control Strategies

To reduce transient errors, one can use a technique called feed forward control. In this case, an accurate system model is used to predict the theoretical control action required for the desired trajectory. This theoretical control action is then superimposed with the existing control action. We will focus on two classical types of feed forward controllers.

3.1.1 Torque Feed Forward (TFF) Controllers

This control strategy is very common for robotic manipulators, where the required control action is time varying and nonlinear [34-37], [56, 57]. It uses an inverted plant model for calculating the torque that the motor needs to provide in order to achieve the desired motion. This calculated torque is added to the position feedback control action of the respective axis. Figure 3.6 shows a block diagram of this controller strategy for a time invariant linear single servo axis.

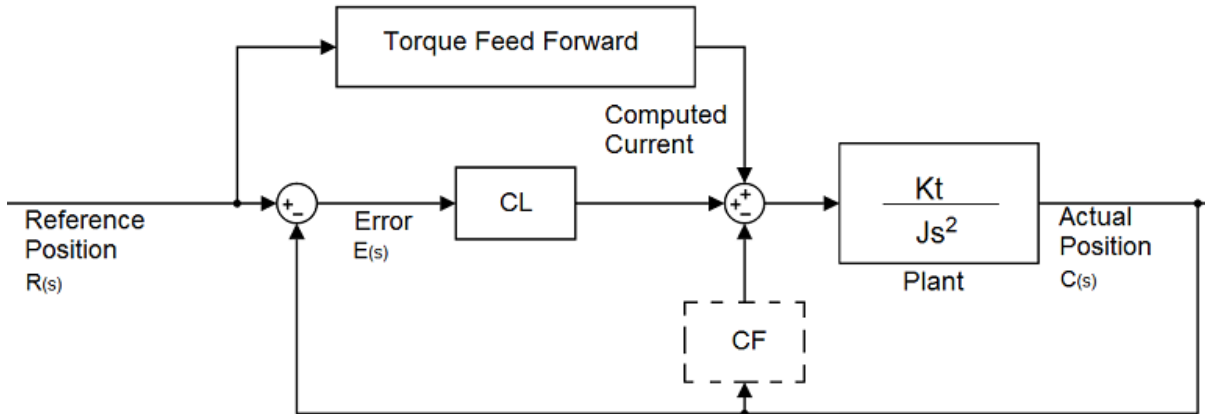


Figure 3. 6 Torque feed forward control block diagram

In Figure 3.6 C_L represents the control loop regulator and C_F is the feedback regulator. To achieve a unity gain control system, the Torque feed forward block would need to be:

$$F_{TFF (unity\ gain)} = \frac{J \times s^2}{K_t} + C_F. \quad (3.6)$$

Both PD and PID controllers do not contain a C_F term, thus the classical Torque feed forward formulation reduces to the inverse of the plant. The SF contains both a C_L and a C_F term. However if Equation (3.6) is applied to SF control, the SF control system is transformed to the PD controller with torque feed forward. Thus torque feed forward will only be applied to PD and PID controllers in this thesis.

It should be noted that plant models that contain unstable zeros in the transfer function numerator cannot be inverted easily without leading to instability. The plant model in this thesis has no zeros and can be inverted easily.

3.1.2 The Zero Phase Error Tracking (ZPET) Controller

The Zero Phase Error Tracking (ZPET) controller, introduced by Tomizoka [25], minimizes the axial error by reducing the phase lag between the actual and the reference position. In contrast to torque feed forward, where an inverted plant model is used, ZPETC employs the inverse of the position loop transfer function as a pre filter (see Figure 3.7). As a result, zeroes and poles of the position loop transfer function will be canceled and there is no lag between the actual and the desired position. Care has to be taken when inverting control loops with zeros on the right hand side of the s plane, since it would lead to unstable poles in the pre filter. However, since the control systems used in this study have only stable zeros, the pre filter is an exact inverse of the position loop transfer function.

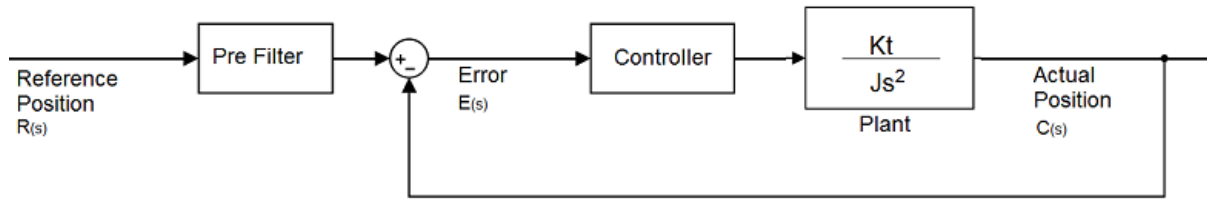


Figure 3. 7 ZPET block diagram

3.2 Cross Coupled Controller (CCC)

Koren [28] noticed that the contour error due to axis mismatch or controller bandwidth limitations in circles changes very slowly along the path. Thus, he proposed to add a contour error compensation scheme to the conventional Cartesian servo control scheme. The suggested method employs axial errors to calculate the contour error. The initial method only considers linear contours as reference trajectories. Figure 3.8 shows a reference point P on a linear reference trajectory, the corresponding actual point P^* , axial errors E_x , E_y , contour error ϵ_c , and tangential error ϵ_T .

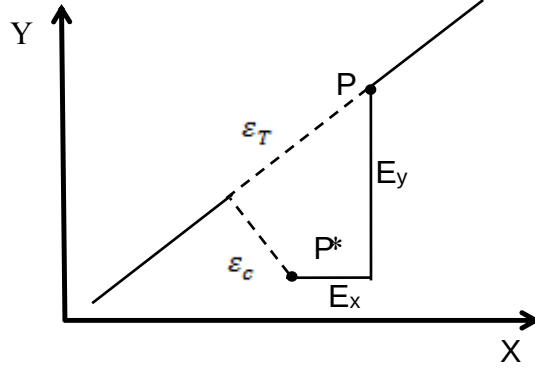


Figure 3. 8 Axial, contour, and tangential errors for a linear contour

The contour error, ε_c is extracted from the axis errors E_x , and E_y using cross coupling gains CC_x and CC_y :

$$\varepsilon_c = -CC_x E_x + CC_y E_y. \quad (3.7)$$

The cross coupling gains are functions of the cutting directions and can be calculated from the axis velocities $\dot{X}$ and $\dot{Y}$:

$$CC_x = \frac{\dot{Y}}{\sqrt{\dot{X}^2 + \dot{Y}^2}} \quad (3.8)$$

$$CC_y = \frac{\dot{X}}{\sqrt{\dot{X}^2 + \dot{Y}^2}}. \quad (3.9)$$

A PID control action is then performed on the contour error. Finally, CC_x and CC_y are employed a second time in order to transfer the control action back to individual axis coordinate frames. Figure 3.9 shows a block diagram of the CCC algorithm.

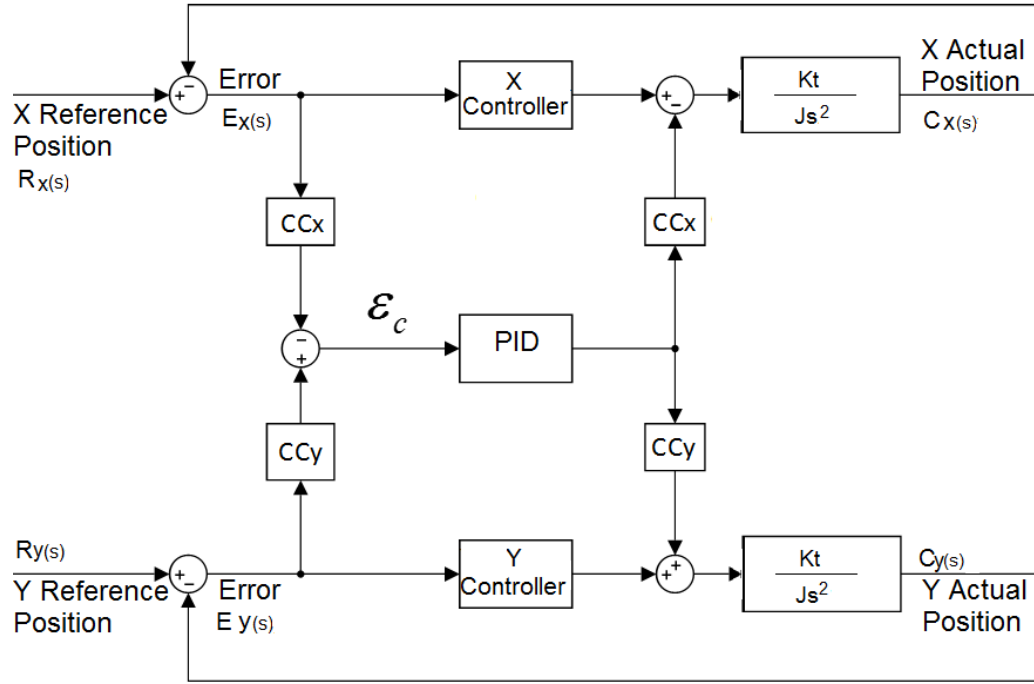


Figure 3. 9 CCC block diagram

It should be noted that the original CCC algorithm was derived for straight lines. Later, the algorithm was updated for lines of arbitrary curvature [Koren 1991]. However this algorithm requires the knowledge of local radius of curvature at the control level. Recently, Rahaman et al. [62] introduced linearized cross coupled control (LCCC), which does not require the local radius of curvature without compromising control accuracy.

3.3 Tangential-Contouring (TC) Controller

Taking Koren's idea to control contour error rather than axis position error one step further, leads to the Tangential-Contouring control strategies that are often also called a Task Coordinate Frame approach (TCFA). This strategy calculates errors in the normal and tangential direction of the trajectory, applies a control law in the normal and tangential direction and then transforms the control actions back into the drive directions. This technique was first formulated by Chiu and Tomizuka [41]. In this thesis, a slightly simpler version of this original work (see Lo and Chung [1999]) is adopted which does not require updating the controller gains with trajectory information. Chiu and Tomizouka [2001]

indicate that this omission would lead to error during fast motions at low curvature. However, this study's experiments indicate that these errors are small.

The TC control scheme is similar to CCC. They both use the same transformations for calculating contour error, and the same PID style normal controller. However, the TC controller replaces the two axis servo controllers with a single tangential controller.

Assuming a linear contour, the TC controller employs Equations (3.10) and (3.11) to calculate the normal and tangential errors. The transformation coefficients CC_x and CC_y are identical to the ones used for CCC shown in Equations (3.8) and (3.9)

$$\varepsilon_c = -CC_x E_x + CC_y E_y \quad \text{Contour error} \quad (3.10)$$

$$\varepsilon_t = CC_y E_x + CC_x E_y \quad \text{Tangential error} \quad (3.11)$$

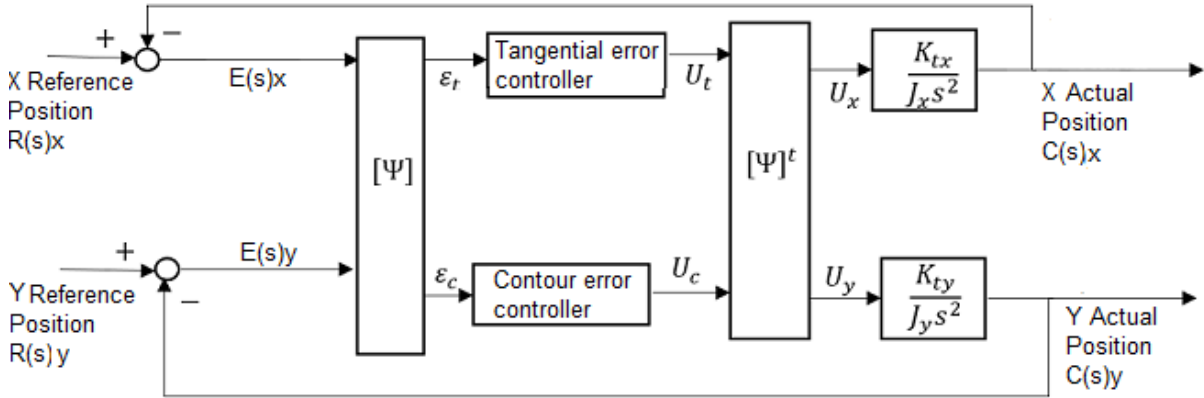


Figure 3. 10 TC block diagram [43]

The block diagram of a TC controller shown in Figure 3.10, indicates that the errors, ε_c and ε_t , are fed to the contour and tangential controllers respectively, and the controllers' actions are then transferred back and fed into the individual axis plants.

3.4 Experimental Comparison of Control Strategies

Two sets of experiments are carried out to determine the best combination of controller strategies. First, circular interpolation is studied. These experiments highlight the strength of contour error control systems such as CCC and TC, since they have relatively slowly changing trajectories in normal directions. In the second set of experiments, corner tracking is investigated in order to compare the contouring performance during rapid transients.

3.4.1 Experimental Setup

The experiments are carried out in the custom built high speed biaxial table shown in Figure 3.11.

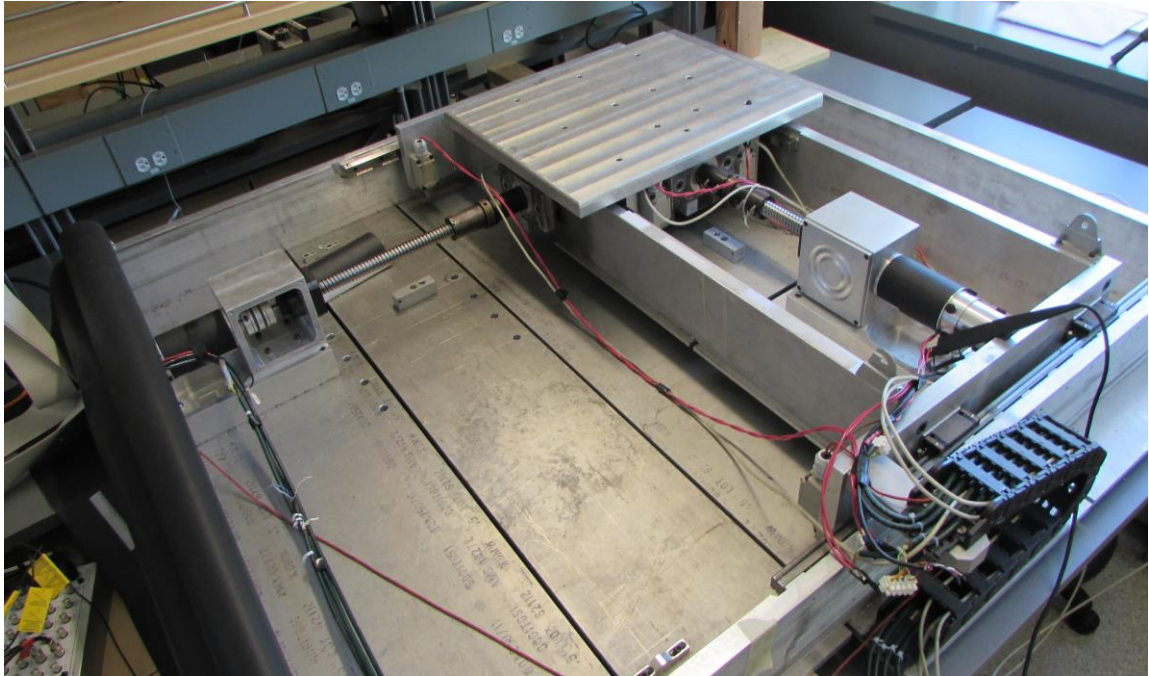


Figure 3. 11 Biaxial table used for the experiments

The two orthogonal axes of the high speed biaxial table are driven by brushed DC motors and ball screws with a pitch of 0.00101 m/rad . In order to determine actual positions of the motors and calculate the axial errors, 1000 line encoders are employed that are attached to

each motor. Copley 311 PWM amplifiers drive the motors in torque mode. Position control methods are implemented in a DSpace 1103 DSP control board. The natural frequency and damping ratio of the controllers are set to the 20 Hz and 1 respectively. The sample update frequency is set to 25 kHz. The parameters of the experimental table are summarized in Table 3.2.

Table 3. 2 Experimental setup

$J_x=4.4205\text{e-}04$	$[\frac{kg}{m^2}]$	Total Inertia of X axis
$J_y=4.7736\text{e-}04$	$[\frac{kg}{m^2}]$	Total Inertia of Y axis
$p_x p_y = 0.00101$	$[\frac{m}{rad}]$	Ball screw pitch
$K_{tx}, K_{ty} = 0.463$	$[Nm / A]$	Motor constant
$R_x, R_y = 4.5$	$[\Omega]$	Motor resistance
$I_{max,x}, I_{max,y} = 5$	$[A]$	Maximum current
$U_{max,x}, U_{max,y} = 110$	$[V]$	Maximum voltage
$\dot{X}_{max} = \dot{Y}_{max} = 0.2$	$[\frac{m}{s}]$	Maximum table speed
$\ddot{X}_{max} = \ddot{Y}_{max} = 5$	$[\frac{m}{s^2}]$	Maximum table acceleration
$\omega_c = 126$	$[rad/sec]$	Controller bandwidth

3.4.2 Circular Interpolation

The parameters of the circular trajectory are summarized in Table 3.3. The feed velocity during circular interpolation is selected at $\frac{1}{2}$ the maximum axis velocity (see Figure 3.13) and the angular frequency of the circular trajectory is selected at approximately $\frac{1}{4}$ of the bandwidth of the control system. Since the frequency content of the trajectories during entry

and exit transients is much higher than during the steady state portion with constant velocity, it is expected that transient errors are larger than the steady state errors. At the beginning and end of the trajectory, axis acceleration is limited to $5 \frac{m}{s^2}$ in order to avoid current saturation of the axis drives (see Figure 3.12).

Table 3. 3 Circular trajectory parameters

$\omega_{circle} = 31.5$	rad/sec	Angular frequency of reference trajectory
$r_{circle} = 3.175$	mm	Radius of the reference trajectory

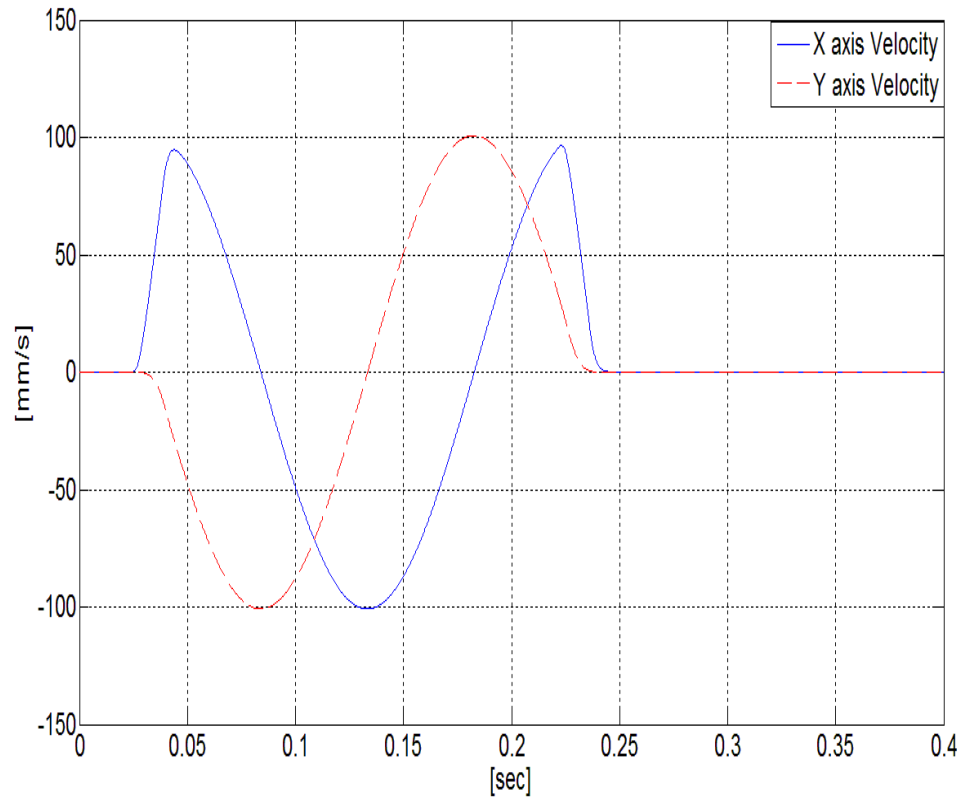


Figure 3. 12 Axes velocities

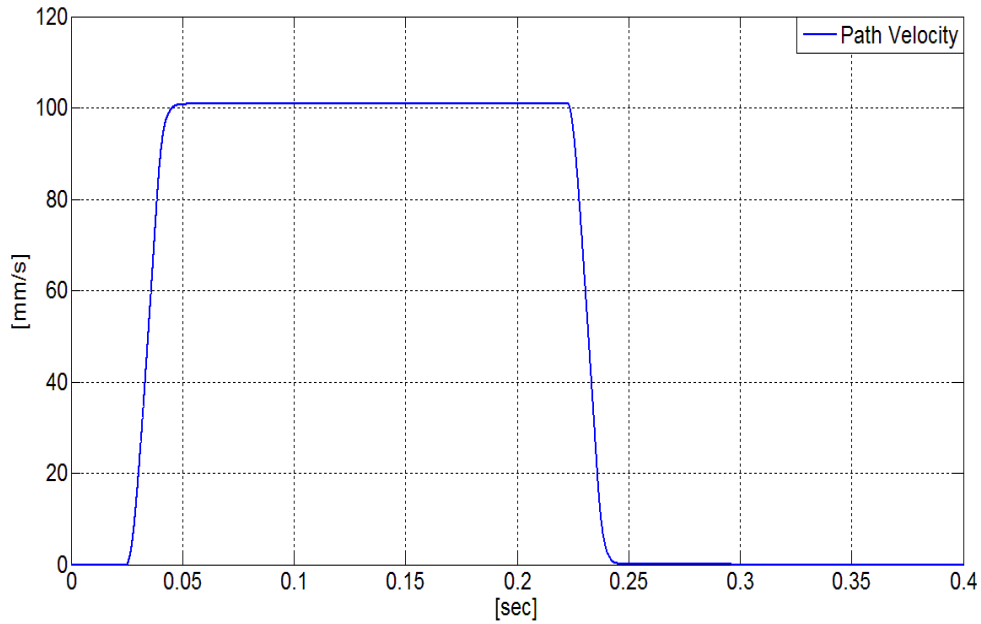


Figure 3. 13 Path (feed) velocity

The results have been divided into two parts. In the first part, no feed forward algorithms are used and CSC using SF, PD, and PID are compared to CCC with SF, and TC control using PD and PID. In the second part, the effects of adding feed forward strategies are studied.

Figure 3.14 compares the contouring errors of control systems that do not rely on model based feed forward terms. The conventional critically damped 2nd order controller (CSC+SF) has up to 300 μ m of steady state contouring error, since the trajectory frequency is too close to the break frequency of the control loop. To reduce this contouring error one can choose a controller with higher order that exhibits a flatter frequency response than the SF controller. Figure 3.14 shows that CSC+PD and CSC+PID controllers with identical roots as the CSC+SF controller do indeed reduce the contouring error compared to the CSC+SF controller. However, the contouring error during the entry transient for the CSC+PID controller is still very large at 100 μ m. One technique to further reduce contouring error is cross coupled control. Both traditional CCC+SF and Linearized Cross Coupled Control (LCCC+SF) add an integral term to the normal direction of the trajectory that effectively eliminates the steady state error of the CSC+SF controller during the circular move. However, transients still show contour errors of 50 μ m. The difference in performance between CCC+SF and LCCC+SF is small and one would usually choose LCCC+SF for its

ease of implementation. Another technique for reducing contouring error is the use of TC control. This technique performs control action in the contour and tangential direction of the trajectory. When combined with PD and PID control schemes, TC+PD provides larger contouring errors than CSC+PD, but TC+PID achieves similar contouring errors as LCCC+SF which is smaller than CSC+PID.

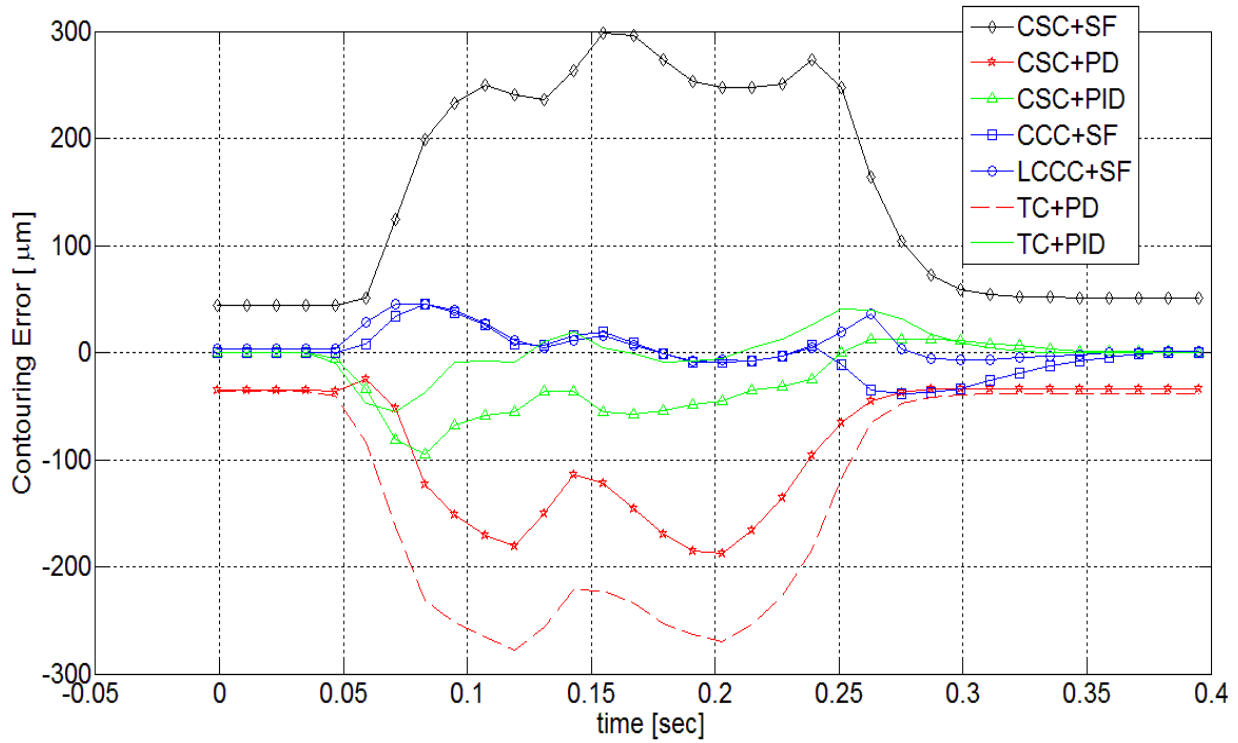


Figure 3. 14 Contouring errors of CSC, LCCC and TC controllers

Inspecting Figure 3.14, one might wonder why there is such a marked improvement, in the steady state contouring error for LCCC+SF and TC+PID compared to conventional PID control. The circular reference trajectory has zero frequency components in the normal direction during the steady state portion of the trajectory. Since both of these algorithms have an integral action in this same normal direction they can compensate for any low frequency disturbances in this direction.

Figure 3.14 also demonstrates, that even higher order controllers (CSC+PID instead of CSC+SF or CSC+PD) cannot follow the complete frequency spectrum of the reference trajectory during entry and exit transients. However, by using feed forward (either ZPETC or

torque feed forward) transient contouring error can largely be mitigated, as long as a good plant model is available. This is demonstrated in Figure 3.15, where various feed forward strategies are compared.

ZPETC has been proposed as a general procedure to achieve low contouring error. In simulations, this technique has zero contouring error when applied to the CSC+SF control system. However, in this experimental study, this finding is not confirmed. In fact, SF+ZPETC delivers larger contouring errors than the CSC+PID scheme without feed forward. These errors are due to un-modelled disturbances like friction in the guides and ball screws. Thus, it becomes apparent, that ZPETC cannot remove errors due to un-modelled disturbances that the original control system does not reject. Disturbance rejection can be improved in two ways. First, one can increase the bandwidth of the controller. Secondly, one can use axis controllers with higher orders (such as ZPETC+CSC+PD or ZPETC+CSC+PID) that provide better disturbance rejection.

Following this logic, ZPETC was tested with PD and PID controllers and significant reductions in transient and steady state contouring errors were observed. As expected, ZPETC+CSC+PID has the smallest errors, and it is able to drive the steady state contouring error at standstill to zero.

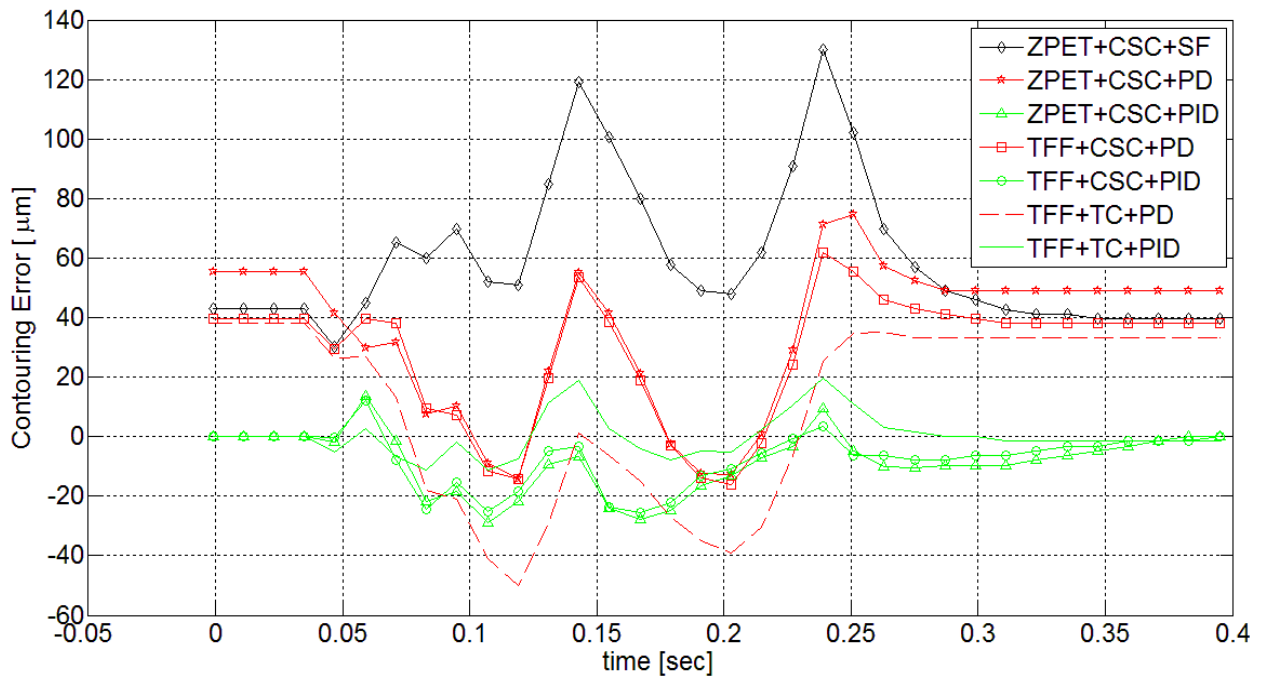


Figure 3. 15 Circular contour errors for ZPET controllers and torque feed forward controllers

An alternative feed forward technique to ZPET is Computed TFF. This technique also provides a unity gain transfer function albeit at a reduced computational cost compared to ZPETC. Figure 3.15 shows that the contouring errors for ZPETC+CSC+PID and TFF+CSC+PID are almost identical. However, marginally better contouring performance is obtained with TFF+TC+PID. Again, the dynamics in the TC coordinate frame are of zero frequency for a circle, and thus, a PID type controller can drive the steady state contouring errors to zero during the steady state portion of the circle.

From these experiments it is concluded that best contouring performance requires good disturbance rejection (PID) to eliminate steady state error, and feed forward control (ZPETC or TFF) in order to eliminate transient error. Surprisingly, the Task Coordinate Frame (TC) approach shows relatively little advantage over the conventional Cartesian Servo Control (CSC). Thus, these experiments suggest, that TFF+CSC+PID should be used as the benchmark for more advanced control studies.

3.4.3 Corner Tracking

Corner tracking between line segments is likely the most common transient machining operation. Thus, it warrants a separate set of experiments. The experimented trajectory (see Figure 3.16), consists of four corners, and four linear segments. During the steady state portion of the trajectory, the feed velocity is set to $\frac{1}{2}$ the maximum axis velocity (see Figure 3.17b).

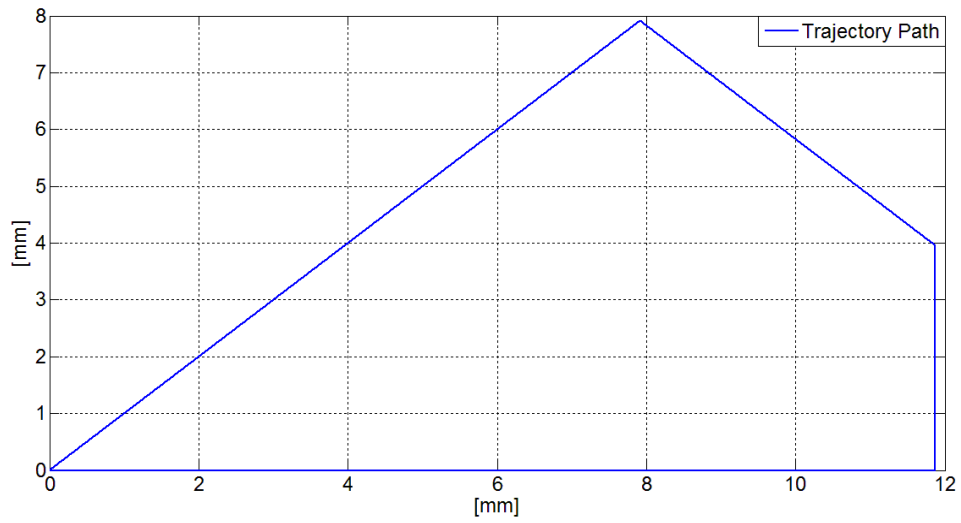


Figure 3. 16 Corner tracking trajectory

Axis acceleration is limited to $5m/s^2$ (see Figure 3.17a) during transients, in order to limit the required axis currents.

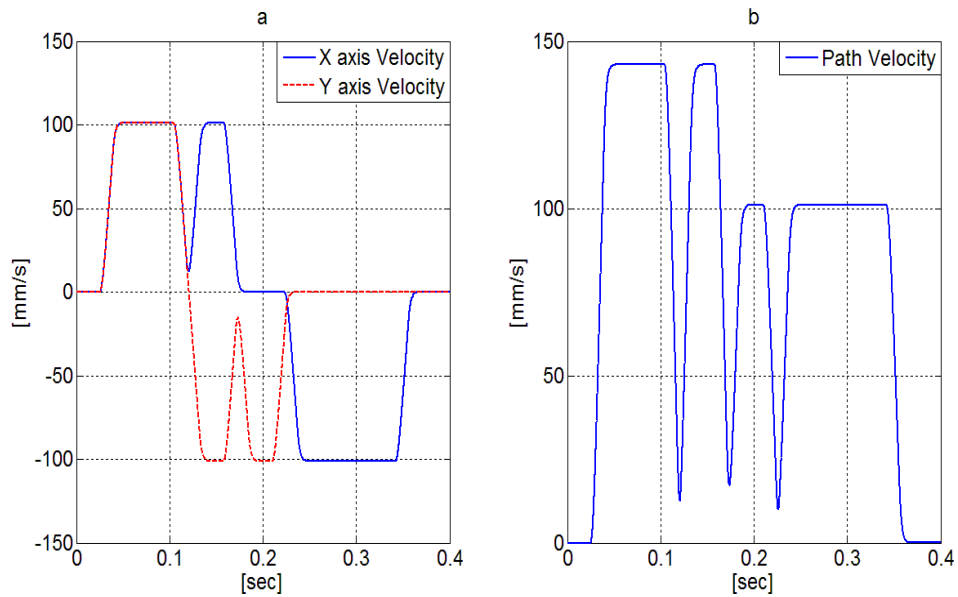


Figure 3. 17 a: Axis velocities. b: Tangential velocity

Figure 3.18, compares the recorded contouring errors of LCCC+SF, TFF+CSC+PID, and TFF+TC+PID for the described corner tracking trajectory.

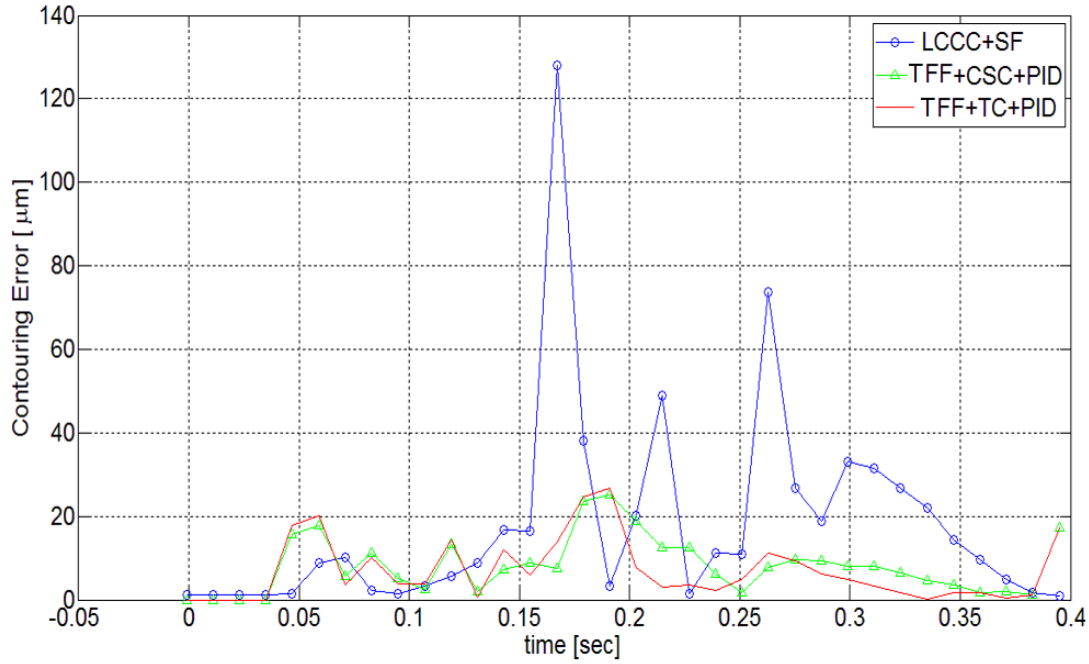


Figure 3.18 Contouring errors for corner tracking

Even though LCCC+SF performs well in steady state and during start and end transients, the contouring errors during the actual corners are well above the desired tolerance of 50μm. This indicates, that this algorithm requires a dwell time between motion segments in order to fully reach its potential. For clarity of Figure 3.18 the original CCC algorithm was not included, since it showed even larger errors in the corners.

There is no considerable difference between the contouring performances of the TFF+CSC+PID, and TFF+TC+PID controllers during corner tracking. The PID control law eliminates steady state errors, and the Torque Feed Forward method strategy reduces transient errors to less than the desired tolerance of 50μm. This confirms the findings from the circular experiments that suggested to use TFF+CSC+PID as a benchmark for advanced control strategies such as fuzzy logic control, robust control, sliding mode control, or model predictive control.

3.5 Discussions

In this chapter, the contouring performance of CSC, CCC, and TC control was benchmarked. Within these three architectures SF, PD and PID controllers were implemented and augmented with feed forward strategies when possible. Both circular interpolation and corner tracking experiments were conducted on a high speed biaxial table.

As expected, PID control outperforms PD and SF control in all tested architectures, since it has superior disturbance rejection throughout the controller bandwidth. Only PID control can achieve zero contouring error during standstill, since the integral term within the PID structure provides infinite disturbance rejection at zero frequency.

LCCC performs well in circles and during start and end transients, but not well in sharp corners. This could be alleviated by introducing dwell times between motion segments, but that would also increase the time required to traverse the path. The smallest contouring errors were achieved with TFF+CSC+PID control and TFF+TC+PID control. In both cases, the PID control action eliminates steady state errors and the FF action reduces transient errors to acceptable levels. TFF+TC+PID control performs slightly better during the steady state portion of a circle than conventional TFF+CSC+PID control. However, the marginal improvement in accuracy requires significantly more computational effort.

Chapter 4: A New Real Time Feedrate Optimization Algorithm for 2-D Cartesian Machine Tools

According to Figure 2.3, the interpolator creates a reference trajectory from the nominal path and feedrate provided by the process planning system. The nominal path is comprised of motion segments that could be lines, circles, or splines. The nominal feedrate limits the velocity along the path. This chapter outlines a new algorithm that will vary the feedrate continuously along the path in order to achieve minimum machining time without violating velocity and acceleration constraints. The algorithm works for any type of path interpolation (lines, circles, splines), since it varies feedrate along the reference path. In addition, the algorithm is computationally inexpensive and can be implemented in real-time.

4.1 Representing Variable Sampling Times in Position Control Systems with Constant Loop Closing Frequencies

To adjust path velocity in real-time at the axis control level, one needs to adjust the rate at which the position increments are processed. Feedrate modulation is a strategy for adjusting the increment processing rate without changing the position loop closing frequency. Feedrate modulation takes advantage of the fact that the position loop bandwidth ($< 40\text{Hz}$) is typically orders of magnitude smaller than the position loop closing frequency (between 1 kHz and 50 kHz). Feedrate modulation inserts zero motion increments for all axes when the feedrate is reduced, and combines several motion increments when the feedrate is increased. Figure 4.1 illustrates two examples of this technique.

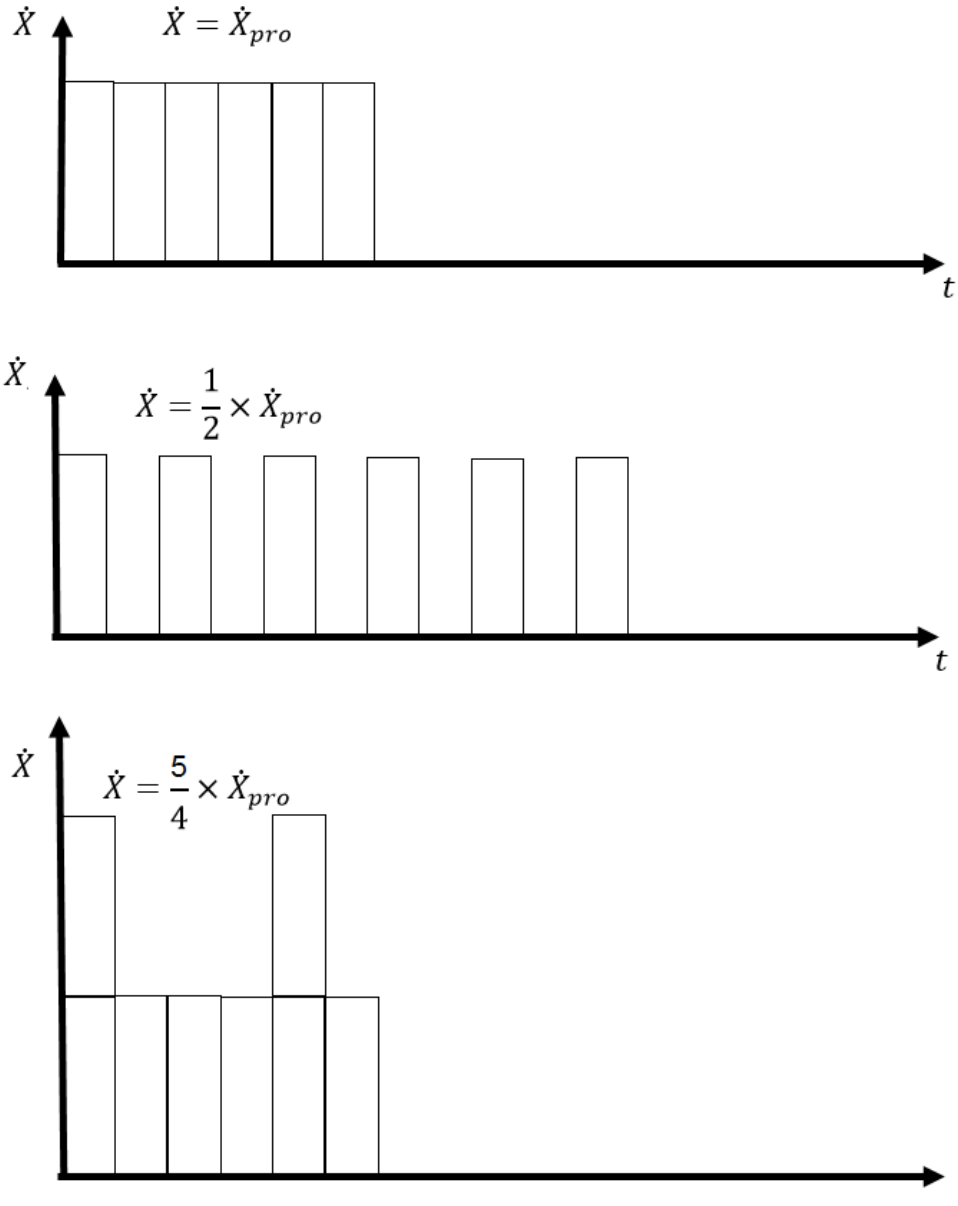


Figure 4. 1 a) X axis moves with programed velocity b) Feedrate modulation by inserting zero motion increments to reduce X axis's velocity to half of the programed velocity c) Combining every fifth sample with its predecessor and 5/4 increase in the programed velocity

Figure 4.1 a) shows motion increments for the X axis without any feedrate modulation. In this scenario, the X axis moves at the programed velocity ($\dot{X}_{pro}$). In Figure 4.1 b) a zero motion increment is added in between each original motion increment. As a result, the X axis slows down to half of its programed velocity. Figure 4.1 c) shows a case where, every

fifth increment is combined with its predecessor, resulting in a 5/4 increase in the axis velocity.

Since the position loop bandwidth is orders of magnitudes smaller than the loop closing frequency, the position controller acts as a low pass filter on the modulated reference trajectory. In a sense, feedrate modulation adjusts feedrate in a fashion similar to how a PWM driver adjusts voltage for a motor drive.

Feedrate modulation has been successfully implemented in the past in order to allow for fast corner tracking [17-20], as well as compensating for machining constraints such as maximum chip thickness constraints [21,22].

The above feedrate modulation algorithms derive an optimum feedrate solely from measured position error. The advantage of this strategy is that it can be implemented with little computational effort and it is robust towards model uncertainties. However, since these strategies only use very limited model knowledge, they can only guarantee feasible but not necessarily optimal trajectories. The new algorithm presented in this chapter uses the reference path in order to arrive at the time optimal feedrate. Measured position error can then be used to account for process constraints such as chip thickness control. However this is outside the scope of this work.

4.2 Expressing Velocity and Acceleration for Sampled Data Systems with Uneven Sample Time

By selectively inserting zero motion increments or combining motion increments, feedrate modulation effectively allows assigning individual sample times to each motion increment. Before developing the algorithm for selecting sample times, expressions for describing velocity and acceleration from motion increments with uneven sample times are required.

Given a string of motion increments ΔX_i the position X_i is defined as:

$$X_i = \sum_{k=0}^i \Delta X_k \quad (4.1)$$

For a system with constant sample time Δt , central difference approximations can be used to obtain velocity $\dot{X}_i$, and acceleration $\ddot{X}_i$:

$$\dot{X}_i = \frac{\Delta X_{i+1} + \Delta X_i}{2\Delta t} \quad (4.2)$$

$$\ddot{X}_i = \frac{\Delta X_{i+1} - \Delta X_i}{\Delta t^2}. \quad (4.3)$$

For adjusting travel velocity, feedrate modulation keeps the motion increments of ΔX_{i+1} and ΔX_i and adjusts individual sample times for each position increment. Thus, the central difference approximations for position and velocity need to be redefined. This is accomplished by fitting a parabola through the sample points:

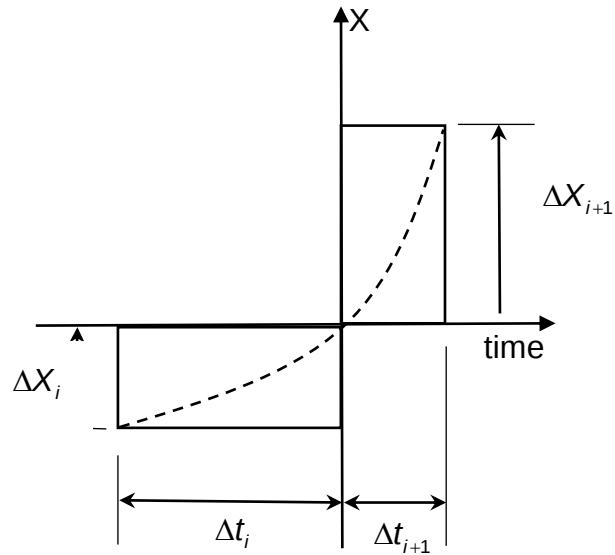


Figure 4. 2 Parabolic position interpolation

$$-\Delta X_i = -\dot{X}_i \Delta t_i + \frac{\ddot{X}_i}{2} \Delta t_i^2 \quad (4.4)$$

$$\Delta X_{i+1} = \dot{X}_i \Delta t_{i+1} + \frac{\ddot{X}_i}{2} \Delta t_{i+1}^2. \quad (4.5)$$

Solving for velocity and acceleration yields:

$$\dot{X}_i = \frac{1}{\Delta t_{i+1} + \Delta t_i} \left(\frac{\Delta X_{i+1}}{\Delta t_{i+1}} \Delta t_i + \frac{\Delta X_i}{\Delta t_i} \Delta t_{i+1} \right) \quad (4.6)$$

$$\ddot{X}_i = \frac{2}{\Delta t_{i+1} + \Delta t_i} \left(\frac{\Delta X_{i+1}}{\Delta t_{i+1}} - \frac{\Delta X_i}{\Delta t_i} \right). \quad (4.7)$$

The attentive reader will notice that Equations (4.6) and (4.7) will reduce to Equations (4.2) and (4.3) when the sample times of two consecutive samples are equal.

4.3 Sample Time Selection for a Single Axis

Before venturing into the case of multiple axes, it is instructive to examine the algorithm for the simpler case of a single X-axis path only. In this case the following constraints need to be met:

$$\dot{X}_i \leq \dot{X}_{max} \quad (4.8)$$

$$\ddot{X}_i \leq \ddot{X}_{max}. \quad (4.9)$$

The time optimal trajectory requires that one of these two constraints is met at all times. A flowchart of the algorithm is shown in Figure 4.3.

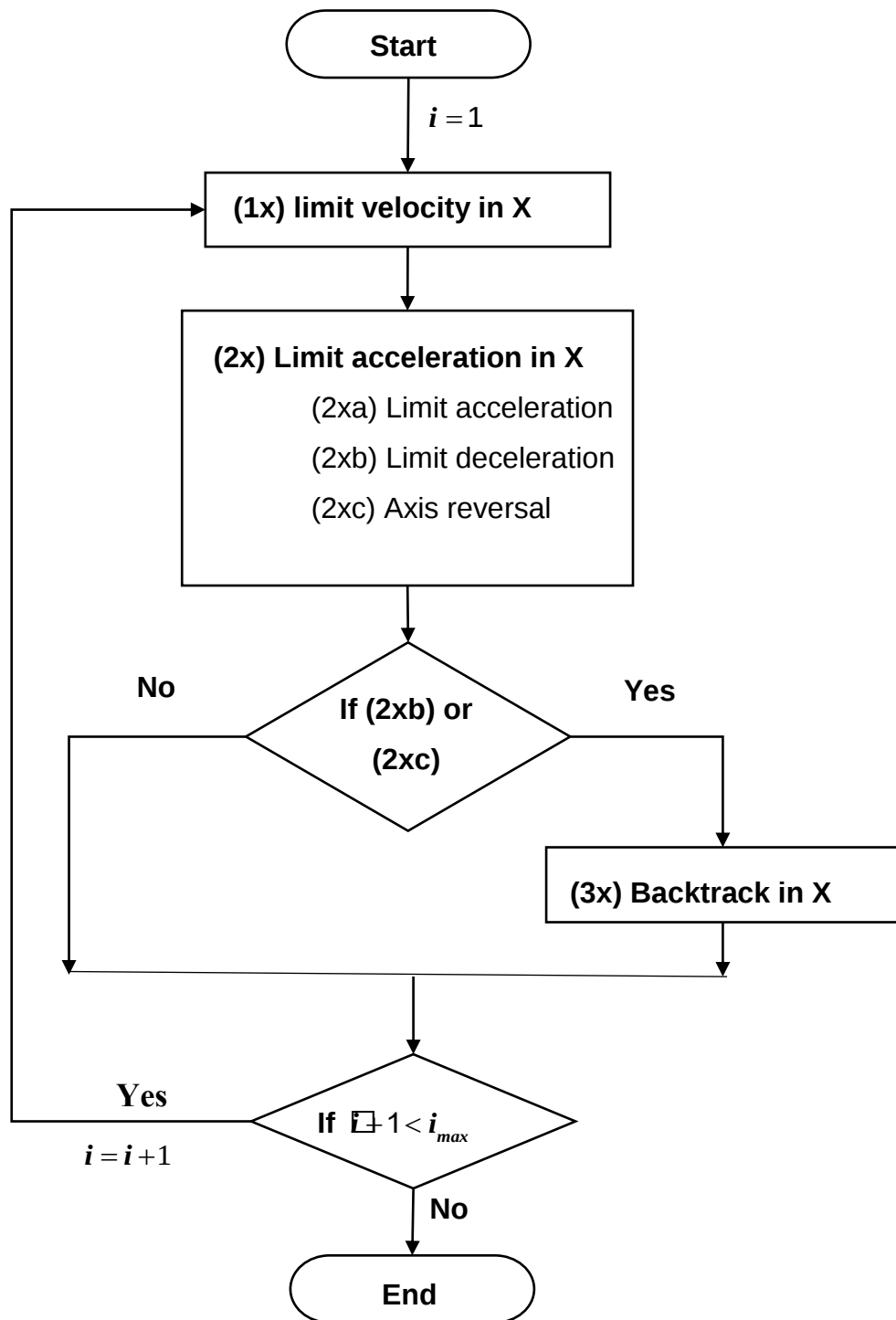


Figure 4. 3 Flowchart of the single axis algorithm

The flowchart shows that the single axis algorithm steps through the motion increments from $i=1$ to $i=i_{\max}$. For every motion increment, velocity (1x) is constrained before acceleration (2x). In case of deceleration (2xb) or reversal (2xc) a final backtracking operation is required. This order of operation is deliberate since it reduces the number of times that the acceleration (2x) and backtracking (3x) operations need to be applied. A more detailed description of the 3 operations is outlined in the following sections.

4.3.1 (1x) Limit Velocity in X

At every motion increment, the velocity constraint is applied first. Figure 4.4 shows a case where the nominal velocity (shown in black) exceeds the allowable maximum velocity (shown in red).

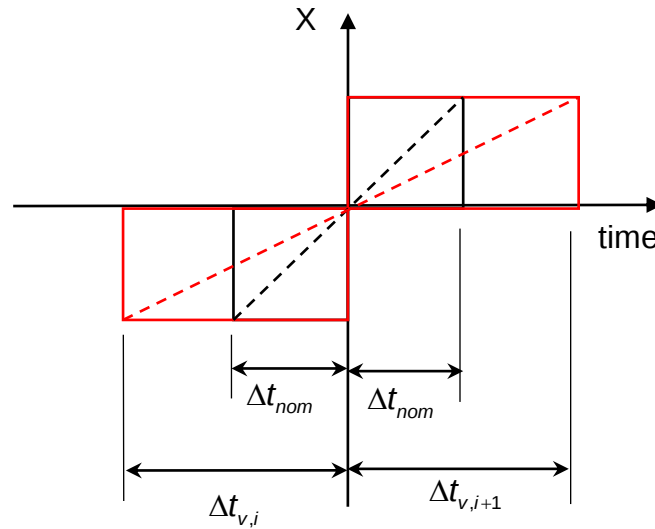


Figure 4. 4 Velocity constraint

It is difficult to apply only Equation (4.6) to constrain velocity, since it is dependent both on Δt_i and Δt_{i+1} . To simplify this task it is noted that:

$$\min\left(\frac{\Delta X_{i+1}}{\Delta t_{i+1}}, \frac{\Delta X_i}{\Delta t_i}\right) < \dot{X}_i < \max\left(\frac{\Delta X_{i+1}}{\Delta t_{i+1}}, \frac{\Delta X_i}{\Delta t_i}\right).$$

This leads to the conclusion that, if both $\frac{\Delta X_{i+1}}{\Delta t_{i+1}}$ and $\frac{\Delta X_i}{\Delta t_i}$ are smaller or equal to $\dot{X}_{max}$, then $\dot{X}_i$ must also be smaller or equal to $\dot{X}_{max}$. Thus, it is sufficient to select sample times on the following basis in order to ensure that Equation (4.8) is observed:

$$\Delta t_{v,i} = \max \left(\Delta t_{nom}, \left| \frac{\Delta X_i}{\dot{X}_{max}} \right| \right) \quad \text{for } i = 1 \quad (4.10a)$$

$$\Delta t_{v,i+1} = \max \left(\Delta t_{nom}, \left| \frac{\Delta X_{i+1}}{\dot{X}_{max}} \right| \right) \quad \text{for } i \geq 1 \quad (4.10b)$$

where, Δt_{nom} is the nominal sample update time of the axis control loop, and $\Delta t_{v,i}$ is the sample time required to meet the velocity constraint. After this step, the velocity constraints are met at all times, but the acceleration still needs to be constrained. To ensure that velocity constraints are not violated in the remaining algorithm, sample times can only be increased during the subsequent operations to limit accelerations.

4.3.2 (2x) Limit Acceleration in X

Three different cases need to be taken into account when constraining acceleration: acceleration, deceleration, and axis reversal.

4.3.2.1 (2xa) Limit acceleration:

Figure 4.5 shows the case where the acceleration resulting from the velocity constraint (1x) shown in black, violates the acceleration constraint.

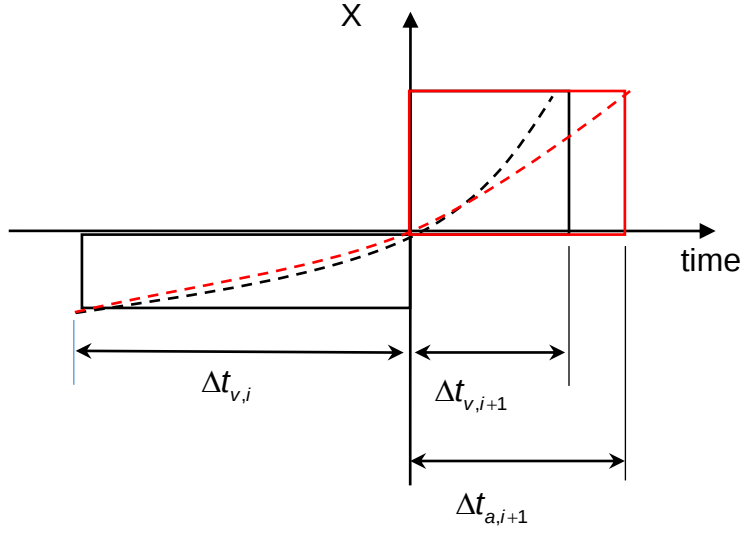


Figure 4. 5 Acceleration constraint

Equation 4.7 indicates that this acceleration can be reduced by decreasing sample time $\Delta t_{v,i}$ or by increasing $\Delta t_{v,i+1}$. However, sample times can only be increased without violating the velocity constraints. Thus, in case of an acceleration violation, $\Delta t_{v,i}$ cannot be reduced, but $\Delta t_{v,i+1}$ needs to be increased using Equation (4.11).

$$\ddot{X}_{max} = |\ddot{X}_i| = \left| \frac{2}{\Delta t_{a,i+1} + \Delta t_{v,i}} \left(\frac{\Delta X_{i+1}}{\Delta t_{a,i+1}} - \frac{\Delta X_i}{\Delta t_{v,i}} \right) \right|. \quad (4.11)$$

It can be shown that, this expression is a second order polynomial in $\Delta t_{a,i+1}$ with only one feasible solution:

$$\Delta t_{a,i+1} = \frac{-\left(\Delta t_{v,i}^2 + \frac{2|\Delta X_i|}{|\ddot{X}_i|} \right) + \sqrt{\left(\Delta t_{v,i}^2 + \frac{2|\Delta X_i|}{|\ddot{X}_i|} \right)^2 + 4 \frac{2|\Delta X_{i+1}| \Delta t_{v,i}^2}{|\ddot{X}_i|}}}{2\Delta t_{v,i}}. \quad (4.12)$$

4.3.2.2 (2xb) Limit Deceleration:

Figure 4.6 Shows the case, where the deceleration resulting from the velocity constraint (1x) shown in black, violates the deceleration constraint:

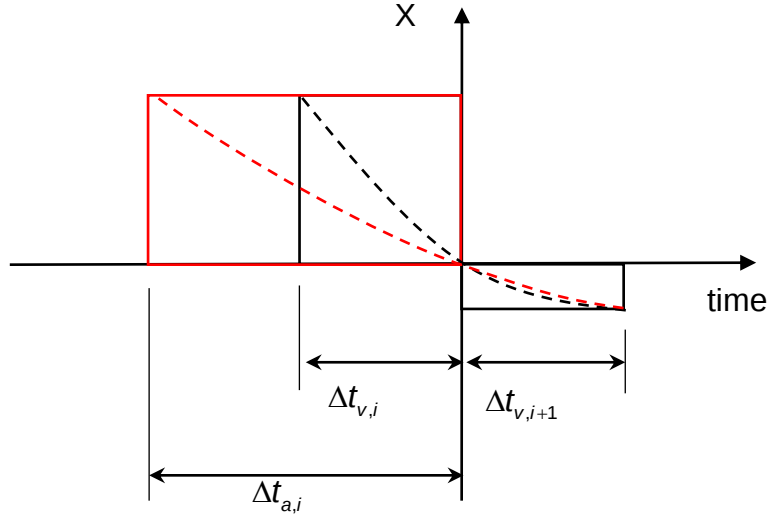


Figure 4. 6 Deceleration constraint

Again, only increased sampling times are allowed and thus, Equation (4.7) can be reformulated to provide a rule for selecting $\Delta t_{a,i}$ in case of a deceleration:

$$\ddot{X}_{max} = |\ddot{X}_i| = \left| \frac{2}{\Delta t_{v,i+1} + \Delta t_{d,i}} \left(\frac{\Delta X_{i+1}}{\Delta t_{v,i+1}} - \frac{\Delta X_i}{\Delta t_{d,i}} \right) \right| \quad (4.13)$$

Using a similar proof for (2xa) it can be shown that there is only one feasible solution of this second order polynomial in $\Delta t_{a,i}$.

$$\Delta t_{d,i} = \frac{-\left(-\Delta t_{v,i+1}^2 - \frac{2|\Delta X_{i+1}|}{|\ddot{X}_i|}\right) + \sqrt{\left(-\Delta t_{v,i+1}^2 + \frac{2|\Delta X_{i+1}|}{|\ddot{X}_i|}\right)^2 + 4\frac{2|\Delta X_i|\Delta t_{v,i+1}^2}{|\ddot{X}_i|}}}{2\Delta t_{v,i+1}} \quad (4.14)$$

It should be pointed out, that an increase in $\Delta t_{v,i}$ can lead to a deceleration violation in step $i-1$, which would require backtracking. This scenario is described after the next section on axis reversal.

4.3.2.3 (2xc) Axis Reversal in X:

Figure 4.7 shows the case, where the acceleration resulting from the velocity constraint (1x), violates the acceleration and deceleration constraint during an X-axis reversal:

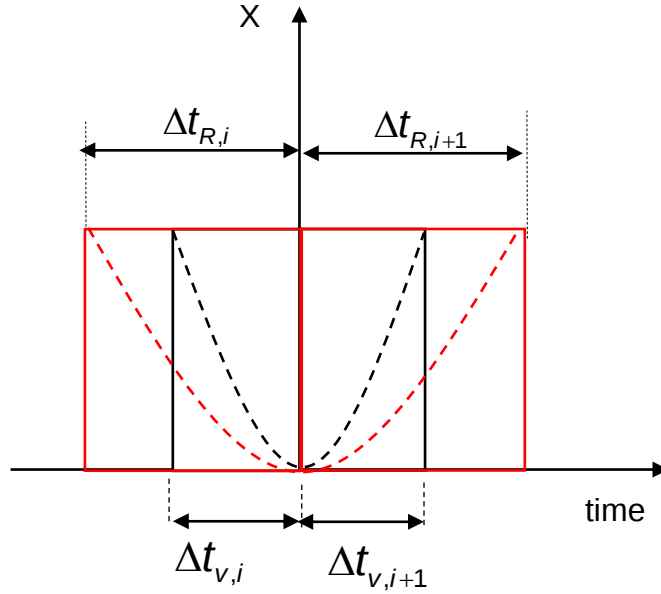


Figure 4. 7 Axis reversal constraint

In this case, step i contains a deceleration and step $i+1$ an acceleration. In order to adjust both constraints at the same time, an additional relationship is required. To achieve continuous feedrate override through this special case, it is assumed that $\Delta t_{R,i} = \Delta t_{R,i+1}$. This leads to the following rearrangement of Equation (4.7)

$$\ddot{X}_{max} = |\ddot{X}_i| = \left| \frac{\Delta X_{i+1} - \Delta X_i}{\Delta t_{R,i}^2} \right| = \left| \frac{\Delta X_{i+1} - \Delta X_i}{\Delta t_{R,i-1}^2} \right|. \quad (4.15)$$

Solving for $\Delta t_{R,i} = \Delta t_{R,i+1}$ yields:

$$\Delta t_{R,i-1} = \Delta t_{R,i} = \sqrt{\frac{|\Delta X_{i+1} - \Delta X_i|}{|\ddot{X}_i|}}. \quad (4.16)$$

The deceleration constraint (2xb) and the reversal constraint (2xc) both increase $\Delta t_{v,i}$. To ensure that decelerations of previous increments are not violated, backtracking is required.

4.3.2.4 (3x) Backtrack in X

If the deceleration constraint (2xb) or the reversal constraint (2xc) result in a deceleration violation of the previous sample, $\ddot{X}_{i-1}$, then the sampling times of up to $j_{max,x}$ motion increments prior to the current motion increment ΔX_i need to be adjusted using the following second order polynomial in $\Delta t_{x,i-j}$:

$$\ddot{X}_{max} = |\ddot{X}_{i-j}| = \left| \frac{2}{\Delta t_{i-j+1} + \Delta t_{x,i-j}} \left(\frac{\Delta X_{i-j+1}}{\Delta t_{i-j+1}} - \frac{\Delta X_{i-j}}{\Delta t_{x,i-j}} \right) \right| \quad \text{for } j = 1 \text{ to } j_{max,x}. \quad (4.17a)$$

Solving (4.17a) for $\Delta t_{x,i-j}$ yields:

$$\Delta t_{x,i-j} = \frac{-\left(-\Delta t_{i-j+1}^2 - \frac{2|\Delta X_{i-j+1}|}{|\ddot{X}_{i-j}|} \right) + \sqrt{\left(-\Delta t_{i-j+1}^2 + \frac{2|\Delta X_{i-j+1}|}{|\ddot{X}_{i-j}|} \right)^2 + 4 \frac{2|\Delta X_{i-j}|\Delta t_{i-j+1}^2}{|\ddot{X}_{i-j}|}}}{2\Delta t_{i-j+1}} \quad (4.17b)$$

$$\Delta t_{b,i-j} = \max(\Delta t_{a,i-j}, \Delta t_{x,i-j}). \quad (4.17c)$$

To reduce the computational effort, backtracking can be aborted as soon as $\Delta t_{x,i-j}$ is smaller than the original sample time $\Delta t_{a,i-j}$.

It is noted that the maximum number of back-steps, $j_{max,x}$ is limited by the number of sample times required for the axis to decelerate from maximum velocity to zero at the maximum deceleration. This scenario is depicted in Figure 4.8 where nominal velocity shown in black is constant at the maximum allowable velocity and switches to zero velocity after n samples. Figure 4.8 also shows the velocity profile after feedrate modulation has been applied (shown in red). The time required to reach zero velocity after feedrate modulation has been applied is denoted as $t_{modulate}$.

Both the nominal and the modulated profiles cover the same distance and can be formulated as:

$$S = \dot{X}_{max} (j_{max,x} \Delta t) = \dot{X}_{max} t_{modulate} - \frac{\ddot{X}_{max}}{2} t_{modulate}^2. \quad (4.18)$$

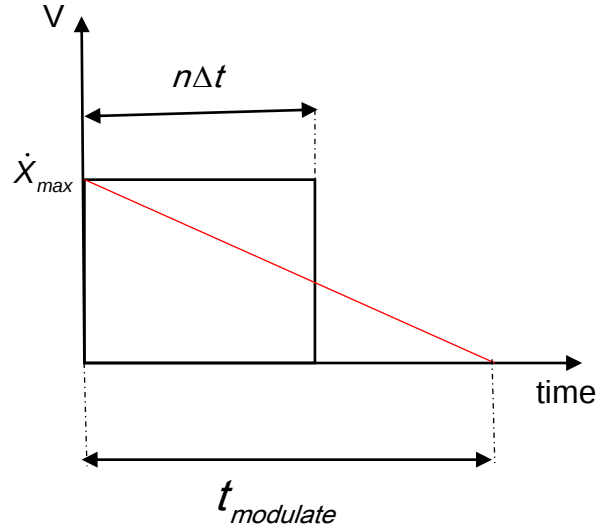


Figure 4. 8 Decelerating from maximum velocity to zero

Also, maximum velocity and maximum allowable deceleration of the axis are related by $t_{modulate}$:

$$\dot{X}_{max} = \ddot{X}_{max} t_{modulate}. \quad (4.19)$$

Combining Equations (4.18) and (4.19) the maximum number of backtracking steps, $j_{max,x}$ is obtained as: $j_{max,x} = \frac{\dot{X}_{max}}{2\ddot{X}_{max}\Delta t}$.

4.4 Algorithm for Sample Time Selection for Two Axes

When adding a second axis to this optimization problem, two additional expressions for the velocity and the acceleration of the second axis are added to the original relationships shown in Equations (4.6) and (4.7):

$$\dot{Y}_i = \frac{1}{\Delta t_{i+1} + \Delta t_i} \left(\frac{\Delta Y_{i+1}}{\Delta t_{i+1}} \Delta t_i + \frac{\Delta Y_i}{\Delta t_i} \Delta t_{i+1} \right), \quad (4.20)$$

$$\ddot{Y}_i = \frac{2}{\Delta t_{i+1} + \Delta t_i} \left(\frac{\Delta Y_{i+1}}{\Delta t_{i+1}} - \frac{\Delta Y_i}{\Delta t_i} \right). \quad (4.21)$$

As a result two additional constraints are required:

$$\dot{Y}_i \leq \dot{Y}_{max}, \quad (4.22)$$

$$\ddot{Y}_i \leq \ddot{Y}_{max}. \quad (4.23)$$

A time optimal trajectory requires that at least one of the four constraints in (4.8), (4.9), (4.22), and (4.23) is met at all times. In most cases, this means that a single constraint of a single axis is dominant. The relationships developed for the single axis case can then be applied to define new sample times. The two axis algorithm follows a similar multi step process as the single axis algorithm, however there is an additional step in the case when two

axes are constraint at the same time. The flowchart in Figure 4.9 shows the procedure for enforcing acceleration and velocity constraints for two axes.

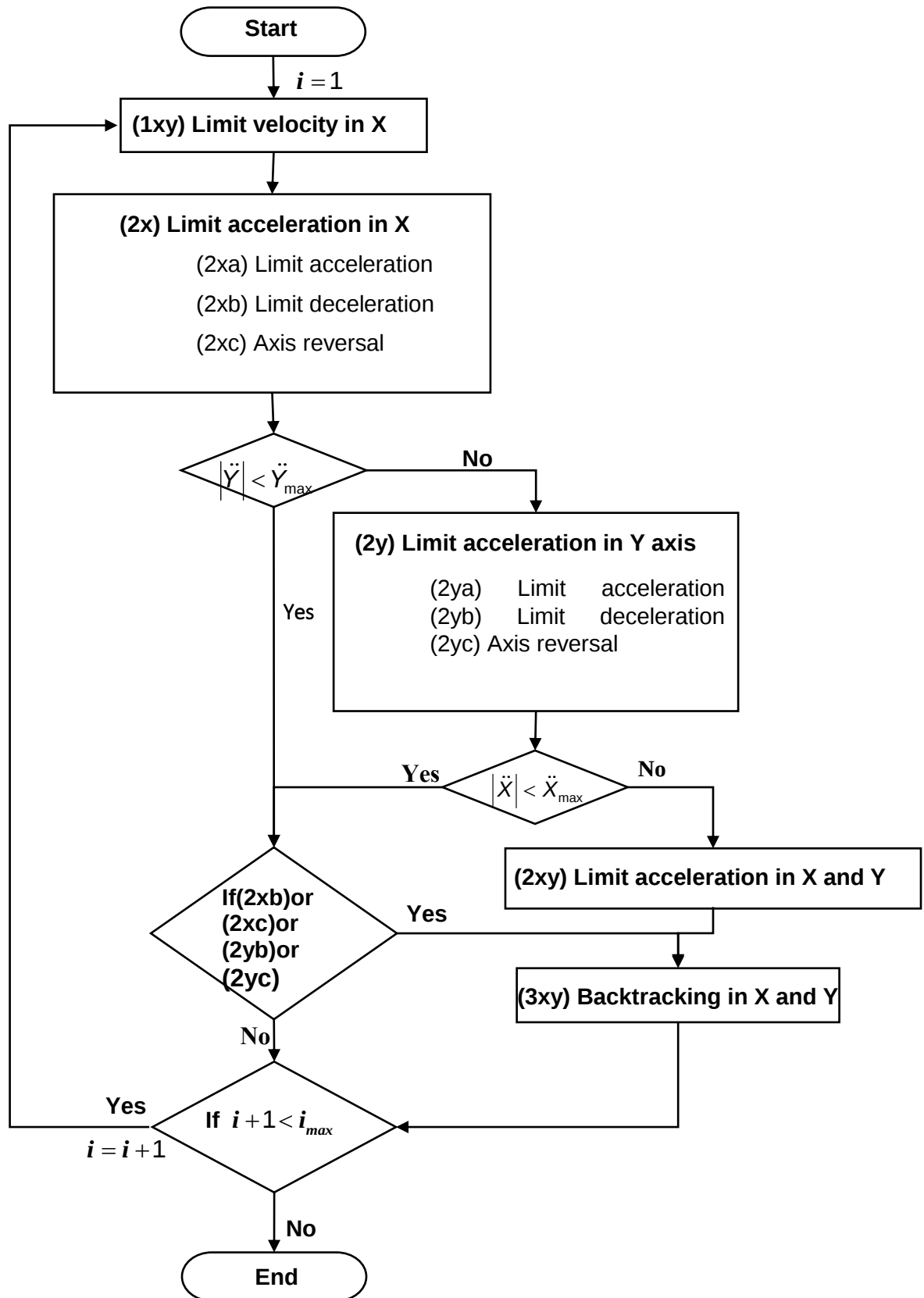


Figure 4. 9 Flowchart of the two axes algorithm

The flowchart shows that the two axis algorithm steps through the motion increments from $i=1$ to $i=i_{\max}$. For every motion increment, velocities in X and Y (1xy) are constrained before the acceleration (2x, 2y, 2xy). In case of deceleration (2xb, 2yb) or reversal (2xc, 2yc) a final backtracking operation (3xy) is required. A more detailed description of this procedure is outlined in the following sections.

4.4.1 (1xy) Limit Velocity in X and Y

All sample times are selected such that velocity is not violated in either of the two axes. The procedure is similar to the single axis case. However in the two axis case, the velocity constraint of the second axis also needs to be taken into account. Extending Equation (4.10) to the two axes case, sample times are defined by:

$$\Delta t_{v,i} = \max \left(\Delta t_{nom}, \left| \frac{\Delta X_i}{\dot{X}_{max}} \right|, \left| \frac{\Delta Y_i}{\dot{Y}_{max}} \right| \right) \quad \text{for } i = 1, \quad (4.24a)$$

$$\Delta t_{v,i+1} = \max \left(\Delta t_{nom}, \left| \frac{\Delta X_{i+1}}{\dot{X}_{max}} \right|, \left| \frac{\Delta Y_{i+1}}{\dot{Y}_{max}} \right| \right) \quad \text{for } i \geq 1. \quad (4.24b)$$

At the end of this step, all the velocity constraints are fulfilled, but acceleration/deceleration constraints still needed to be addressed.

4.4.2 (2x) Limit Acceleration in X, or (2y) Limit Acceleration in Y

After limiting the velocity, accelerations need to be constrained. In many cases one axis represents the dominant constraint. Thus, single axis constraint operations (2x, 2y) can be applied. Only if these operations (2x, 2y) do not yield the desired result, do both axes need to be constrained simultaneously (2xy). This procedure is described in the next section.

4.4.2.1 (2xy) Limit Acceleration in X and Y Axes

This is a case where it is not possible to constrain the acceleration of a single axis without violating the acceleration of the other axis. One such example is shown in Figure 4.10 where the sample times resulting from the velocity constraint (1xy) shown in black, violate the acceleration constraint in the X axis but not in the Y axis. Reducing the violated X axis acceleration to the maximum acceleration by increasing $\Delta t_{v,i+1}$, would result in a violated deceleration in the Y axis.

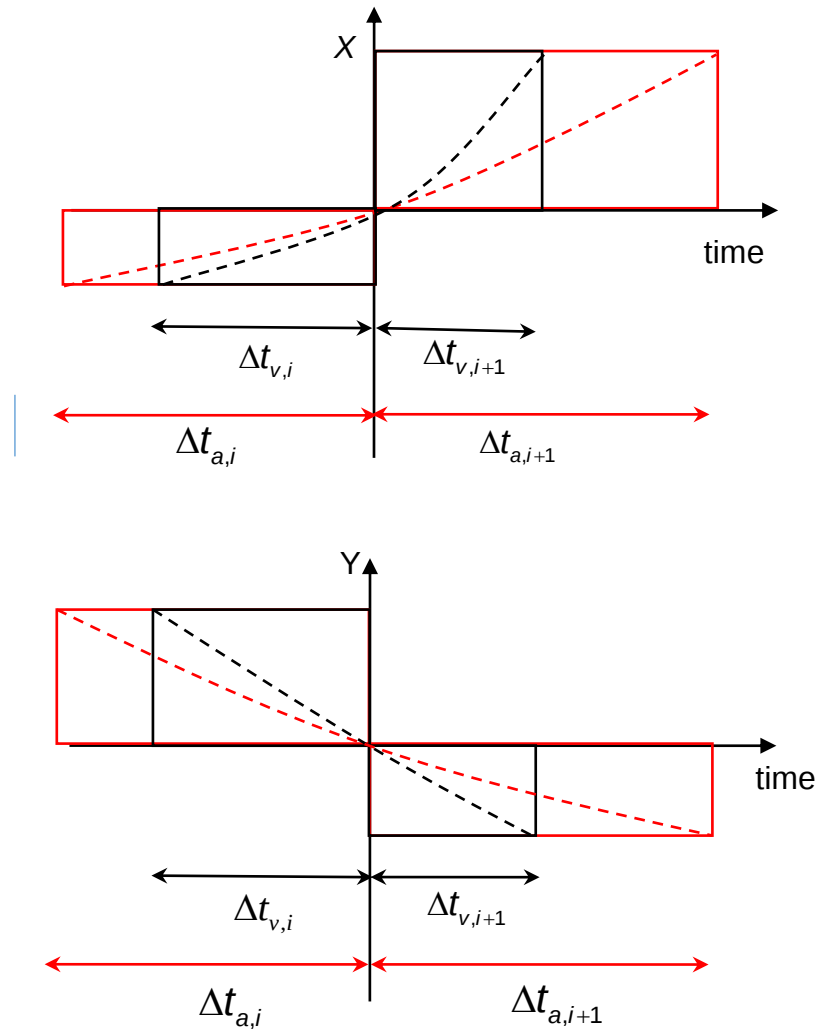


Figure 4. 10 Violated acceleration in X and constrained acceleration in Y (black lines). Simultaneously limiting acceleration in X and Y axes by increasing the sample times (red lines)

Thus, both $\Delta t_{v,i}$ and $\Delta t_{v,i+1}$ need to be increased in order to meet X and Y axis acceleration/deceleration constraints. This is achieved by simultaneously solving the acceleration constraint equations for both axes (Equations (4.25) and (4.26)). Again, these are second order polynomials with only one feasible solution.

$$\ddot{X}_{max} = |\ddot{X}_i| = \left| \frac{2}{\Delta t_{a,i+1} + \Delta t_{a,i}} \left(\frac{\Delta X_{i+1}}{\Delta t_{a,i+1}} - \frac{\Delta X_i}{\Delta t_{a,i}} \right) \right|, \quad (4.25)$$

$$\ddot{Y}_{max} = |\ddot{Y}_i| = \left| \frac{2}{\Delta t_{a,i+1} + \Delta t_{a,i}} \left(\frac{\Delta Y_{i+1}}{\Delta t_{a,i+1}} - \frac{\Delta Y_i}{\Delta t_{a,i}} \right) \right|. \quad (4.26)$$

The solution depends on the sign of the X and Y velocities. If the X and Y velocities have the same sign:

$$\Delta t_{a,i+1} = \sqrt{\frac{(\Delta X_{i+1} / \ddot{X}_{max} + \Delta Y_{i+1} / \ddot{Y}_{max}) \left(-(\Delta X_{i+1} / \ddot{X}_{max}) (\Delta Y_i / \ddot{Y}_{max}) + (\Delta Y_{i+1} / \ddot{Y}_{max}) (\Delta X_i / \ddot{X}_{max}) \right)}{(\Delta X_i / \ddot{X}_{max})^2 + 2 \left(\frac{\Delta Y_i}{\ddot{Y}_{max}} \right) \left(\frac{\Delta X_i}{\ddot{X}_{max}} \right) + (\Delta X_i / \ddot{Y}_{max})^2 + \left(\frac{\Delta X_{i+1}}{\ddot{X}_{max}} \right) \left(\frac{\Delta Y_i}{\ddot{Y}_{max}} \right) + \left(\frac{\Delta Y_{i+1}}{\ddot{Y}_{max}} \right) \left(\frac{\Delta X_i}{\ddot{X}_{max}} \right) + \left(\frac{\Delta Y_{i+1}}{\ddot{Y}_{max}} \right) \left(\frac{\Delta y_i}{\ddot{Y}_{max}} \right) + \left(\frac{\Delta Y_{i+1}}{\ddot{X}_{max}} \right) \left(\frac{\Delta Y_i}{\ddot{X}_{max}} \right)}}, \quad (4.27a)$$

$$\Delta t_{a,i} = \Delta t_{a,i+1} \left| \frac{\frac{\Delta X_i}{\ddot{X}_{max}} + \frac{\Delta Y_i}{\ddot{Y}_{max}}}{\frac{\Delta X_{i+1}}{\ddot{X}_{max}} + \frac{\Delta Y_{i+1}}{\ddot{Y}_{max}}} \right|. \quad (4.27b)$$

If the X and Y velocities have different signs:

$$\Delta t_{a,i+1} = \sqrt{\frac{(\Delta X_{i+1} / \ddot{X}_{max} + \Delta Y_{i+1} / \ddot{Y}_{max}) \left((\Delta X_{i+1} / \ddot{X}_{max}) (\Delta Y_i / \ddot{Y}_{max}) - (\Delta Y_{i+1} / \ddot{Y}_{max}) (\Delta X_i / \ddot{X}_{max}) \right)}{(\Delta X_i / \ddot{X}_{max})^2 - 2 \left(\frac{\Delta Y_i}{\ddot{Y}_{max}} \right) \left(\frac{\Delta X_i}{\ddot{X}_{max}} \right) + (\Delta X_i / \ddot{Y}_{max})^2 - \left(\frac{\Delta X_{i+1}}{\ddot{X}_{max}} \right) \left(\frac{\Delta Y_i}{\ddot{Y}_{max}} \right) + \left(\frac{\Delta Y_{i+1}}{\ddot{Y}_{max}} \right) \left(\frac{\Delta X_i}{\ddot{X}_{max}} \right) + \left(\frac{\Delta Y_{i+1}}{\ddot{Y}_{max}} \right) \left(\frac{\Delta y_i}{\ddot{Y}_{max}} \right) + \left(\frac{\Delta Y_{i+1}}{\ddot{X}_{max}} \right) \left(\frac{\Delta Y_i}{\ddot{X}_{max}} \right)}}, \quad (4.28a)$$

$$\Delta t_{a,i} = \Delta t_{a,i+1} \left| \frac{\frac{\Delta X_i}{\ddot{X}_{max}} - \frac{\Delta Y_i}{\ddot{Y}_{max}}}{\frac{\Delta X_{i+1}}{\ddot{X}_{max}} - \frac{\Delta Y_{i+1}}{\ddot{Y}_{max}}} \right|. \quad (4.28b)$$

One might ask whether there is always a solution to this second order equation that has increased sample times for both axes (a requirement in order to guarantee that velocity constraints are not violated). This can be proven with a conceptual two step thought process. For the example shown in Figure 4.10, increasing $\Delta t_{v,i+1}$ reduces acceleration in x and increases deceleration in y. By simply increasing $\Delta t_{v,i+1}$ until the acceleration constraint in x is not violated, a deceleration constraint in Y would be violated. Thus, the first step of the proof requires that $\Delta t_{v,i+1}$ be increased to $\Delta t_{s1,i+1}$ until $\frac{\ddot{X}_{s1}}{\ddot{Y}_{s1}} = \frac{\ddot{X}_{max}}{\ddot{Y}_{max}}$. At this point, the acceleration constraint in x and the deceleration constraint in y are both violated. However, in the second step of the proof we can now increase both $\Delta t_{s1,i+1}$ and $\Delta t_{v,i}$ by the same

$$\text{factor } SF = \sqrt{\frac{\ddot{X}_{s1}}{\ddot{X}_{max}}} = \sqrt{\frac{\ddot{Y}_{s1}}{\ddot{Y}_{max}}} :$$

$$\begin{aligned} \ddot{X}_{s2} &= \frac{2}{SF \Delta t_{s1,i+1} + SF \Delta t_{v,i}} \left(\frac{\Delta X_{i+1}}{SF \Delta t_{s1,i+1}} - \frac{\Delta X_i}{SF \Delta t_{v,i}} \right) = \\ &= \frac{2}{\Delta t_{s1,i+1} + \Delta t_{v,i}} \left(\frac{\Delta X_{i+1}}{\Delta t_{s1,i+1}} - \frac{\Delta X_i}{\Delta t_{v,i}} \right) \frac{1}{SF^2} = \ddot{X}_{max}, \end{aligned} \quad (4.29)$$

$$\begin{aligned} \ddot{Y}_{s2} &= \frac{2}{SF \Delta t_{s1,i+1} + SF \Delta t_{v,i}} \left(\frac{\Delta Y_{i+1}}{SF \Delta t_{s1,i+1}} - \frac{\Delta Y_i}{SF \Delta t_{v,i}} \right) = \\ &= \frac{2}{\Delta t_{s1,i+1} + \Delta t_{v,i}} \left(\frac{\Delta Y_{i+1}}{\Delta t_{s1,i+1}} - \frac{\Delta Y_i}{\Delta t_{v,i}} \right) \frac{1}{SF^2} = \ddot{Y}_{max}. \end{aligned} \quad (4.30)$$

Thus, during both steps of the proof sample times were increased, and we ended up with a solution in which both X and Y axis constraints are simultaneously fulfilled.

4.4.2.2 (5) Backtracking in X and Y

If the deceleration constraint in 2x, 2y, or 2xy results in a deceleration violation of the previous sample, $\ddot{X}_{i-1}$ then the sampling times of up to $j_{\max,xy}$ motion increments prior to the current motion increment $\Delta X_i, \Delta Y_i$ need to be adjusted. Similar to the single axis scenario, the algorithm recursively adjusts previous sample times $\Delta t_{a,i-j}$ using the following equations:

$$\ddot{X}_{\max} = |\ddot{X}_{i-j}| = \left| \frac{2}{\Delta t_{i-j+1} + \Delta t_{x,i-j}} \left(\frac{\Delta X_{i-j+1}}{\Delta t_{i-j+1}} - \frac{\Delta X_{i-j}}{\Delta t_{x,i-j}} \right) \right| \quad \text{for } j = 1 \text{ to } j_{\max,xy}, \quad (4.31a)$$

$$\ddot{Y}_{\max} = |\ddot{Y}_{i-j}| = \left| \frac{2}{\Delta t_{i-j+1} + \Delta t_{y,i-j}} \left(\frac{\Delta Y_{i-j+1}}{\Delta t_{i-j+1}} - \frac{\Delta Y_{i-j}}{\Delta t_{y,i-j}} \right) \right| \quad \text{for } j = 1 \text{ to } j_{\max,xy}, \quad (4.31b)$$

$$\Delta t_{b,i-j} = \max(\Delta t_{a,i-j}, \Delta t_{x,i-j}, \Delta t_{y,i-j}). \quad (4.31c)$$

At every time increment there is a dominant axis that will require a longer time increment. If both Equations (4.31a) and (4.31b) result in shorter time increments $\Delta t_{x,i-j}, \Delta t_{y,i-j}$ than the original increment $\Delta t_{a,i-j}$, then the backtracking operation can be aborted. The maximum number of back steps $j_{\max,xy}$ is defined by the following equation:

$$j_{\max,xy} = \max \left(\frac{\dot{X}_{\max}}{2\ddot{X}_{\max}\Delta t}, \frac{\dot{Y}_{\max}}{2\ddot{Y}_{\max}\Delta t} \right). \quad (4.32)$$

One might ask whether the solution for $\Delta t_{b,i-j}$ can always be obtained by increasing the time increment $\Delta t_{a,i-j}$ (a requirement in order to guarantee that velocity constraints are not violated). This can be proven using an upper and lower bound for $\Delta t_{b,i-j}$. The lower bound

for $\Delta t_{b,i-j}$ is $\Delta t_{a,i-j}$, where acceleration constraints are violated due to the application of 2x,2y,or2xy in step i :

$$\left| \ddot{X}_{i-1} \right|_{\Delta t_{b,i-1}=\Delta t_{a,i-1}} > \ddot{X}_{max} \quad (4.33)$$

or

$$\left| \ddot{Y}_{i-1} \right|_{\Delta t_{b,i-1}=\Delta t_{a,i-1}} > \ddot{Y}_{max} . \quad (4.34)$$

The upper bound for $\Delta t_{b,i-j}$ is infinity, where $\ddot{X}_{i-1}$ and $\ddot{Y}_{i-1}$ are both zero:

$$\lim_{\Delta t_{b,i-1} \rightarrow \infty} \ddot{X}_{i-1} = \frac{2}{\Delta t_{b,i-1} + \Delta t_{v,i}} \left(\frac{\Delta X_i}{\Delta t_{v,i}} - \frac{\Delta X_{i-1}}{\Delta t_{b,i-1}} \right) = 0, \quad (4.35a)$$

$$\lim_{\Delta t_{b,i-1} \rightarrow \infty} \ddot{Y}_{i-1} = \frac{2}{\Delta t_{b,i-1} + \Delta t_{v,i}} \left(\frac{\Delta Y_i}{\Delta t_{v,i}} - \frac{\Delta Y_{i-1}}{\Delta t_{b,i-1}} \right) = 0. \quad (4.35b)$$

Thus, there must be a solution between these two bounds in which the accelerations of at least one of the two axes must be equal to their constraint value:

$$\Delta t_{a,i-1} < \Delta t_{b,i-1} < \infty, \begin{cases} \left| \ddot{X}_{b,i-1} \right| \leq \ddot{X}_{max} \\ \left| \ddot{Y}_{b,i-1} \right| \leq \ddot{Y}_{max} \end{cases} . \quad (4.36)$$

4.5 Experimental Setup

The experiments to validate the feedrate optimization algorithm are carried out on the same biaxial table as in Chapter 3. The experimental setup is outlined in Section 3.5.

In this chapter, we employ Cartesian Servo Control with Torque Feedforward and PID position control (TFF+CSC+PID). Figure 4.11 shows the controller structure of a single axis of the table.

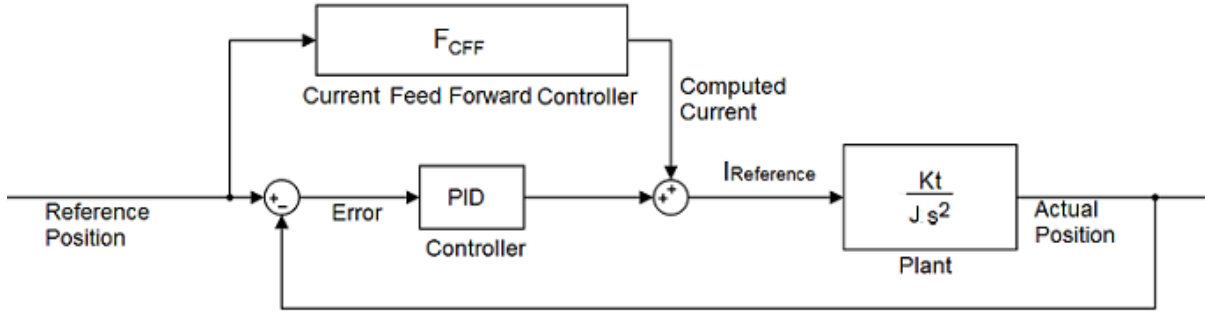


Figure 4. 11 Control system

In order to better show different limit cases for the feedrate optimization algorithm, the controller bandwidth and the maximum allowable values for velocity and acceleration have been altered from the ones in Chapter 3. The new values are shown in Table 4.1.

Table 4. 1 Experimental setup

$\dot{X}_{max}, \dot{Y}_{max} = 0.12$	$\left[\frac{m}{s} \right]$	Maximum table speed
$\ddot{X}_{max}, \ddot{Y}_{max} = 3.68$	$\left[\frac{m}{s^2} \right]$	Maximum table acceleration
$\omega_c = 94$	$\left[\frac{rad}{sec} \right]$	Controller bandwidth

4.6 Experimental Results

A teardrop curve, which is shown in Figure 4.12 is chosen for the experimental validation of the feedrate modulation algorithm.

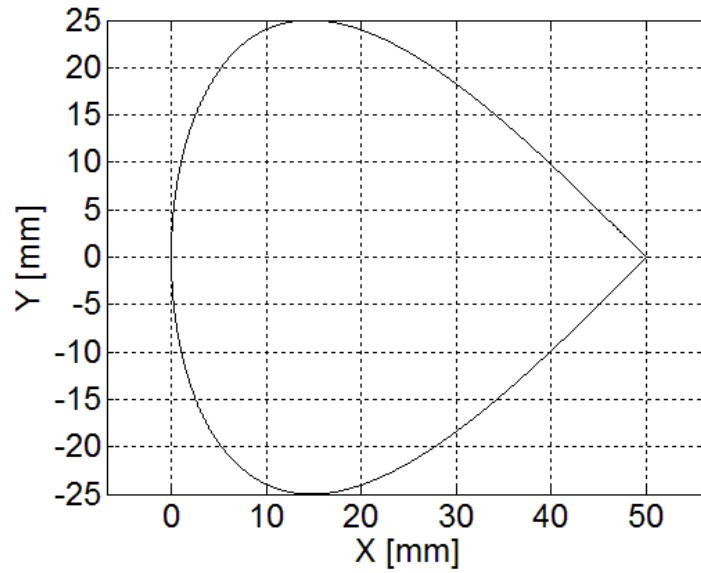


Figure 4. 12 A teardrop curve is used as the original reference path

Figure 4.13 shows the original reference position, original reference velocity, and original reference acceleration profiles of the tear drop. The velocity in both axes are violated during most parts of the original reference path. The motion starts with an acceptable accelertaion in X and violated accelertaion in the Y axis. In the middle of the path, there is a velocity reversal in the X direction, which leads to a violated deceleration in the X axis. The path ends with an acceptable deceleration in the X axis and a violated deceleration in the Y axis. Having different combinations of acceleration/deceleration cases, this path is a good case study to examine the performance of the proposed algorithm.

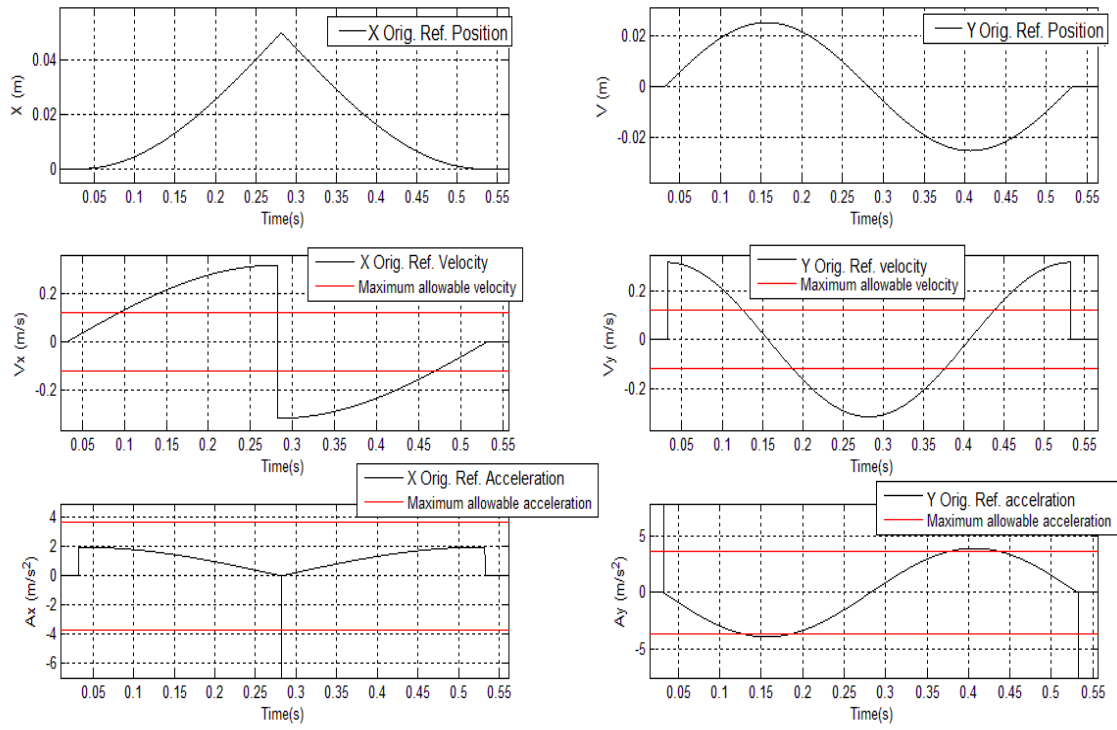


Figure 4. 13 Original reference path, velocity and acceleration profiles of the teardrop

The proposed algorithm shown in Figure 4.9 is used to modulate feedrate and provide feasible reference velocity and acceleration profiles. This optimized feedrate is then fed to the biaxial table. The resulting measurements of velocity and acceleration are plotted together with the reference velocities and accelerations in Figures 4.14 and 4.15 respectively.

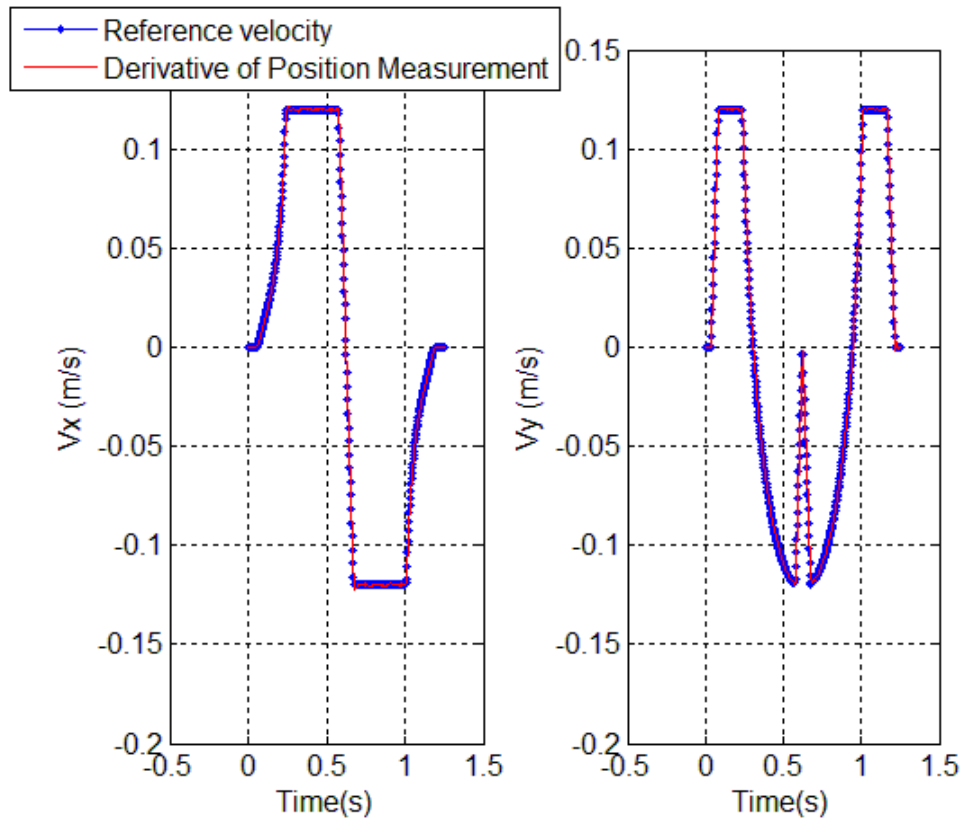


Figure 4. 14 Reference and actual velocity

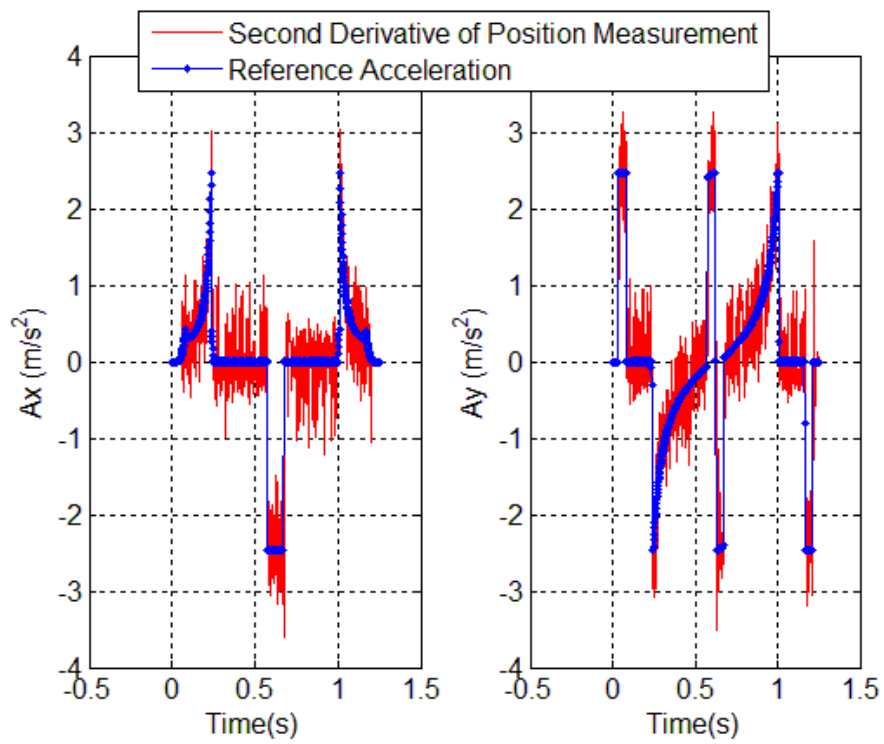


Figure 4. 15 Reference and actual acceleration

Reference velocity and accelerations are shown in blue dotted lines. Velocities are constrained to 0.12m/s and accelerations to 3.68m/s^2 . The velocities and accelerations obtained by taking numerical derivatives of the position measurements are shown in solid red and they follow the desired profiles closely.

An attentive reader will notice that, at any time instant at least one axis moves either at its maximum velocity or maximum acceleration/deceleration. In other words, the algorithm successfully has created a time optimal feedrate for the biaxial table. One important point of interest is the corner of the teardrop on the far right side of Figure 4.14, where the high speed table has to decrease its velocity to zero at the corner point and increase velocity after the corner. Reference and measured velocities during the corner tracking can be seen in Figure 4.14 between time $0.57(\text{sec})$ and $0.64(\text{sec})$. Figure 4.16 provides a detailed view of the reference and measured acceleration/decelerations at the corner point.

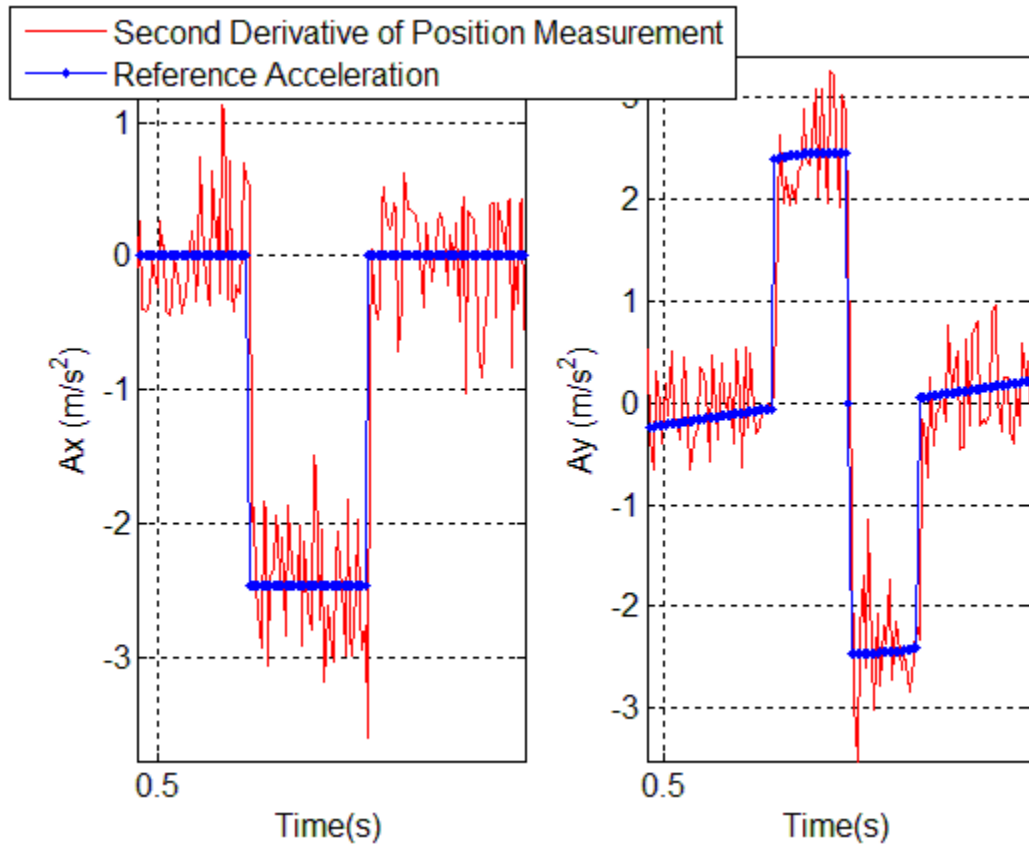


Figure 4. 16 Reference and actual acceleration at the corner of the trajectory

During corner tracking, the X axis moves at its maximum deceleration, while the Y axis changes from maximum acceleration to the maximum deceleration. This corresponds to the simultaneous limitation of acceleration in X and Y performed by block 2xy in Figure 4.9. Again, the measured acceleration closely follows the reference acceleration.

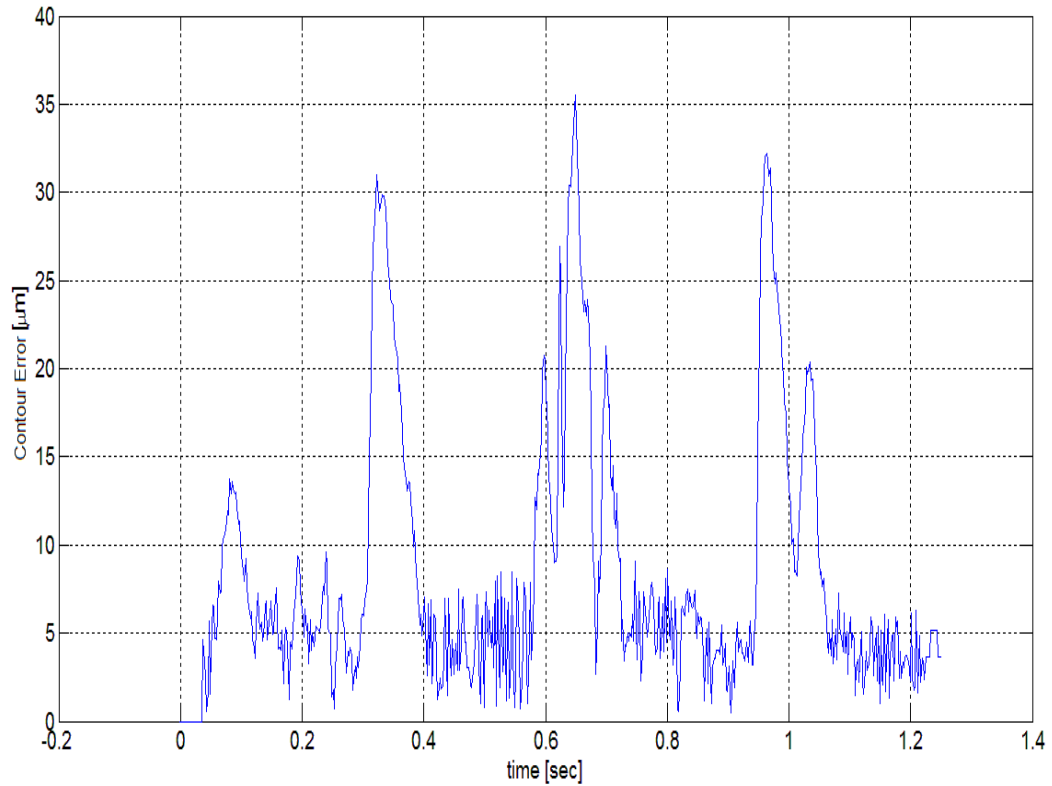


Figure 4. 17 Contour error

Figure 4.17 shows the contour error of the biaxial table. During the velocity constrained portions of the trajectory, contour errors are close to the resolution of the position sensors at around $5 \mu\text{m}$. During the acceleration constraint portion of the trajectory, transient contour errors stay below $35 \mu\text{m}$, which represents a very low value given the extreme accelerations encountered.

4.7 Discussions

In this chapter of the thesis a new real time feedrate optimization algorithm is proposed to generate a guaranteed time optimal feedrate for high speed biaxial CNC tables in the presence of velocity and acceleration constraints. The proposed algorithm is verified on a biaxial table using a teardrop path. Experimental results show that the algorithm successfully limits the different violated velocity/acceleration/deceleration cases and provides the machine with a time optimal feedrate. The main advantage of the proposed algorithm is that it is independent of the trajectory representation, which makes it suitable for a vast variety of machining processes. Also, in contrast to work published in [16], the algorithm requires only a finite look ahead window instead of the whole trajectory, which makes it applicable for real time implementations.

Chapter 5: Reference Trajectories with Tolerance Bounds

The interpolator receives a nominal path and feedrate from the process planning system and converts them to a reference trajectory for the control system (see Figure 2.3). Conventional interpolators will generate a reference path that exactly corresponds to the nominal path. To avoid current saturation at critical points with large normal accelerations (such as corners), the reference velocity needs to be reduced substantially. In the case of an actual corner, the velocity needs to be reduced to zero in order to minimize contouring error. This strategy is inefficient because it does not take machining tolerances into account. This chapter proposes two new interpolation techniques which utilize a predefined tolerance bound around the nominal path to generate a reference path within the tolerance bounds. As a consequence, the machine tool will not need to fully stop at critical points of the path. The first proposed algorithm is limited to corners and the second algorithm is a general algorithm applicable to any type of path.

5.1 Tolerance Bounds at Corner Points

Figure 5.1 shows a sharp corner which consists of two straight lines. It also shows tolerance bounds (shown in red) around the straight portions of the tool path, which are extended to their intersection at the corner. Also, the tolerance bound at the corner point is shown as a tolerance circle (shown in red). The actual trajectory can lie within the coinciding area of tolerance bounds, and tends to go around the corner on a curved path. Hence, a maximum allowable radius can be defined for the path through the corner [70]. In this study, this radius is defined as the radius of the tangent circle to the tolerance circle.

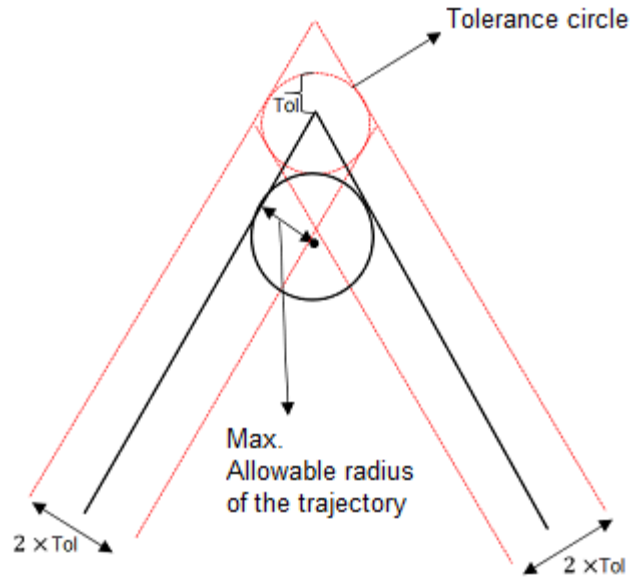


Figure 5. 1 Tolerance boundary lines and tolerance circle (shown in red) around the tool path (shown in black) [70]

In Figure 5.1, *Tol* is the value of the predefined tolerance. In this chapter, the reference path is modified in accordance with the predefined tolerances.

5.2 Case Studies

When traversing across the corner in Figure 5.1, one could choose a path along the inner tolerance bound, the outer tolerance bound, or a combination of the two bounds. In order to select a good strategy for utilizing the tolerance bounds most effectively, it is instructive to study three special cases: circles, reversals, and arbitrary corners.

5.2.1 Circles

When traversing a circle within a tolerance band, one can either traverse along the inner or the outer tolerance bound (see Figure 5.2):

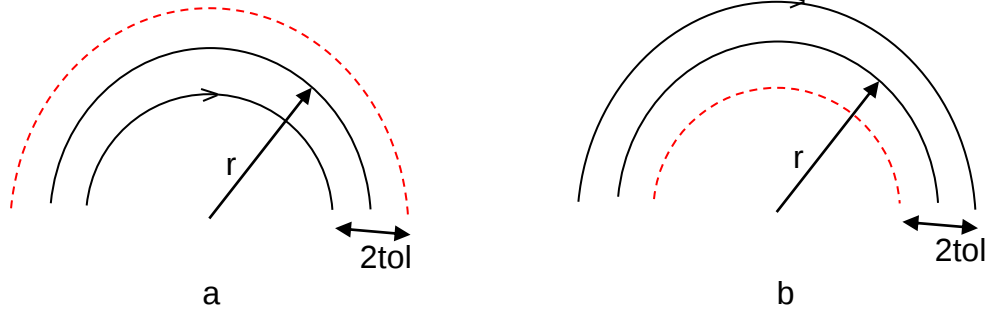


Figure 5. 2 : Traversing along a circle a) along the inner tolerance band b) along the outer tolerance

To determine which strategy leads to faster traverse times, we assume that the tool travels at constant speed as governed by the maximum allowable normal acceleration, $a_{max,normal}$:

$$v_{circle} = \sqrt{ra_{max,normal}}. \quad (5.1)$$

Allowable velocities increase with radius, but the distance travelled also increases with radius. The time to traverse the half circle shown in Figure 5.2 then is:

$$t_{circle} = \frac{r\pi}{v_{circle}} = \pi \sqrt{\frac{r}{a_{max,normal}}}. \quad (5.2)$$

Clearly the traverse time increases when the circle increases with radius, and one would normally aim to traverse the circle along the inner tolerance band. One might ask what the sensitivity of reducing the circle radius is. This is presented in Equation (5.3):

$$\left(\frac{dt}{dr} \right) / \frac{t}{r} = -\frac{1}{2}. \quad (5.3)$$

This indicates that reducing the radius to the inner tolerance limit provides more benefit for small circles.

5.2.2 Single Axis Reversal:

Another interesting special case is the reversal of a single axis travelling at maximum velocity. Two potential strategies come to mind for this scenario. First, one could remain on the programmed path, but reverse a distance equal to the tolerance limit away from the programmed reversal (see Figure 5.3 a). When starting with a velocity v_{max} and decelerating to zero velocity at a rate of a_{max} , the complete reversal would take:

$$t_{rev,a} = \frac{2v_{max}}{a_{max}}. \quad (5.4)$$

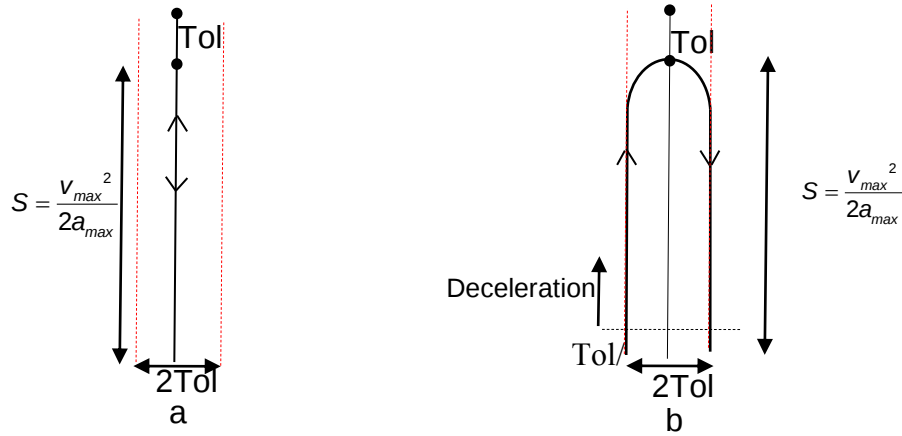


Figure 5. 3 Single axis reversal a) along programmed path b) along outside tolerance band

The second strategy leads along the outside tolerance band as shown in Figure 5.3 b. In this case, the axis does not need to come to a complete stop at the reversal point, but can travel around the circle through the endpoint at a speed of:

$$v_{circle} = \sqrt{a_{max}Tol}. \quad (5.5)$$

Thus, the axis can start to decelerate a distance $n/2$ later than for case a. The resulting reversal time then computes to:

$$t_{rev,b} = t_{rev,a} + (\pi - 2) \sqrt{Tol/a_{max}} + Tol/v_{max}. \quad (5.6)$$

Surprisingly, it is always faster to come to a complete halt than to travel around the outside tolerance bound.

5.2.3 General Corner

For a general corner, two strategies are possible in order to take advantage of the tolerance bands. In the first case, an arc is fit into the inside of the corner, taking advantage of the inside tolerance bound only. This strategy is depicted in Figure 5.4 a where the arc is tangent to the tolerance circle. When the inside angle of the corner is small, then the velocity around the arc needs to be reduced in order to guarantee allowable normal accelerations. In the second case, the actual path approaches the corner on the outside tolerance band, then crosses over the nominal path and passes through the inside angle of the corner before crossing the nominal path after the corner to end up on the outside tolerance band after the corner. This scenario is shown in Figure 5.4 b where the fitted arc is tangent to the tolerance circle.

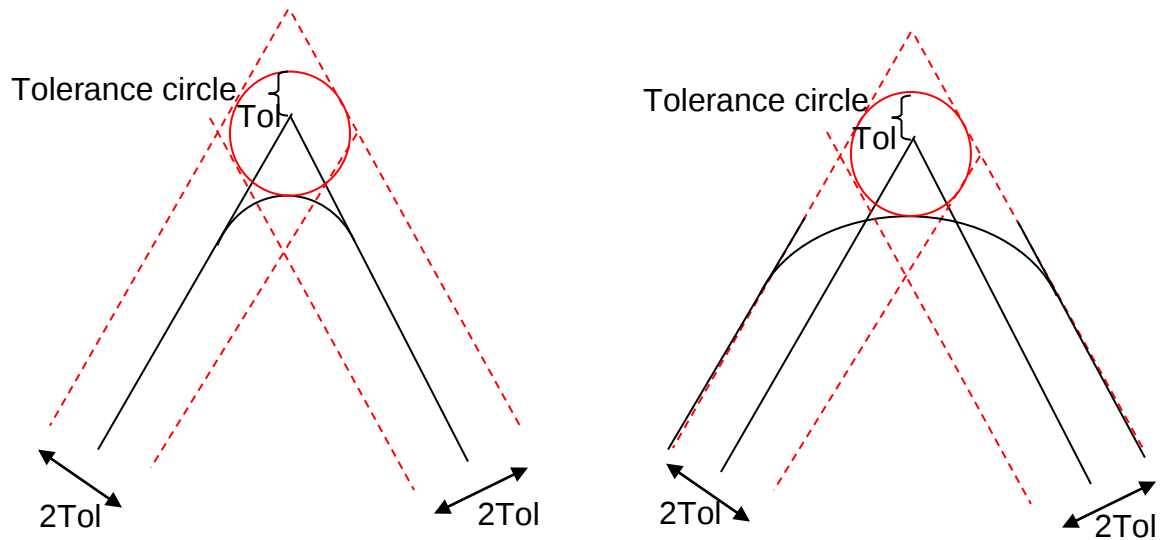


Figure 5. 4 Corner tracking a) along the inside tolerance band, b) between outside and inside tolerance band

The general formulas for these scenarios do not provide much physical insight, and a case study for a typical high speed XY-table is more instructive. Let us consider a maximum velocity of 0.2m/s , a maximum acceleration of 10 m/s^2 , a tolerance band of $50\mu\text{m}$, and a distance of 3.2 mm before and after the corner. Figure 5.5 shows the required travel times for this scenario:

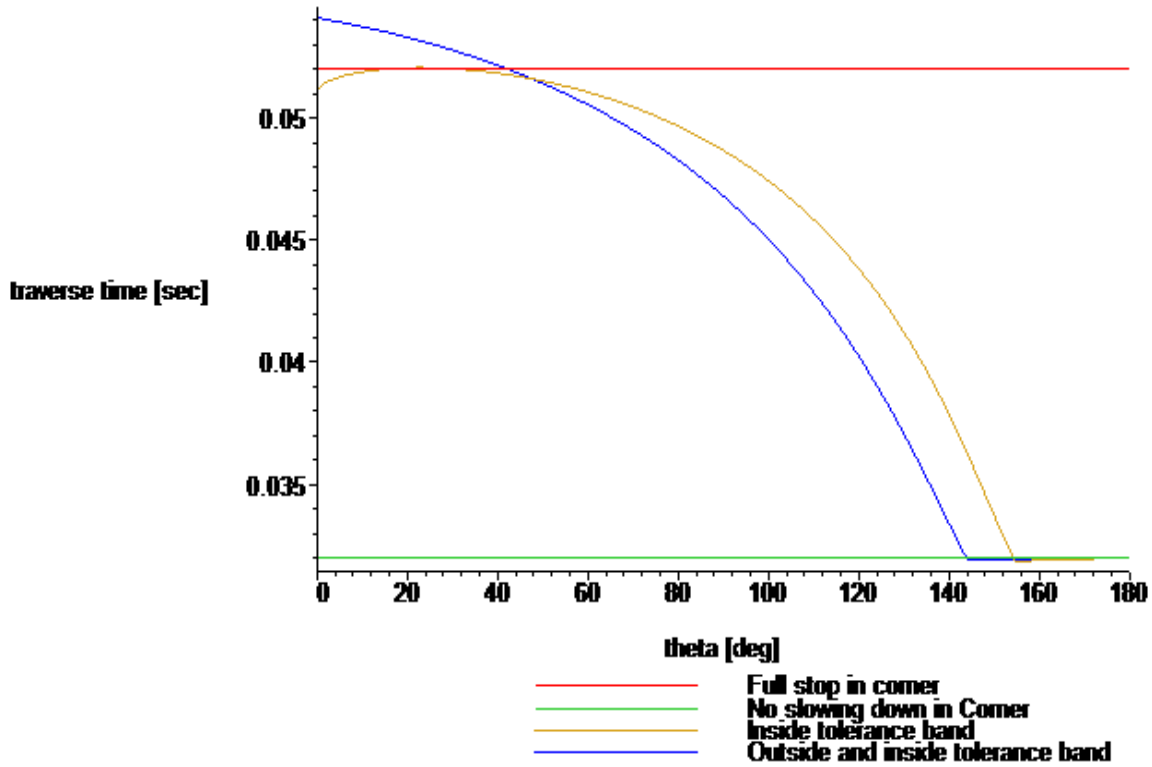


Figure 5. 5 Travel time for a general corner with inside angle theta

The single axis reversal discussed in the previous section has an angle of zero. For this case, Figure 5.5 confirms that it is faster to follow the inside tolerance bound rather than the outside one. However this is not true in general. For large angles, the outside tolerance bound requires less time to traverse the corner. The actual crossover point for selecting the optimum strategy is dependent on the tolerance levels and the programmed travel speed. For a real-time implementation, the strategy following the inside tolerance bound is much easier to compute and still provides significant savings. In the next two sections, two strategies which create trajectories inside the corner are outlined. An experimental comparison of the presented strategies is also provided.

5.3 Fitting an Arc in the Corner

The first strategy (submitted in the ASME IDETC/CIE 2015) is limited to corners. It modifies the nominal path by fitting an arc in the corners of the path without violating the

tolerance limits. As indicated in Figure 5.6 the algorithm requires that the minimum distance between the fitted arc and the corner is less than the tolerance limit, the normal acceleration along the arc is smaller than the maximum allowable acceleration, and the velocity along the arc is set at a constant value which needs to be smaller than the programmed velocity. If the velocity along the arc is smaller than the programmed velocity, then the line segments before and after the arc are used to decelerate and accelerate the tool at maximum allowable acceleration levels in order to match the velocities between the arc and the lines. These rules not only allow for path modification, but they also govern feedrate selection. Thus, the arc fitting algorithm does not require the feedrate selection algorithm described in Chapter 4.

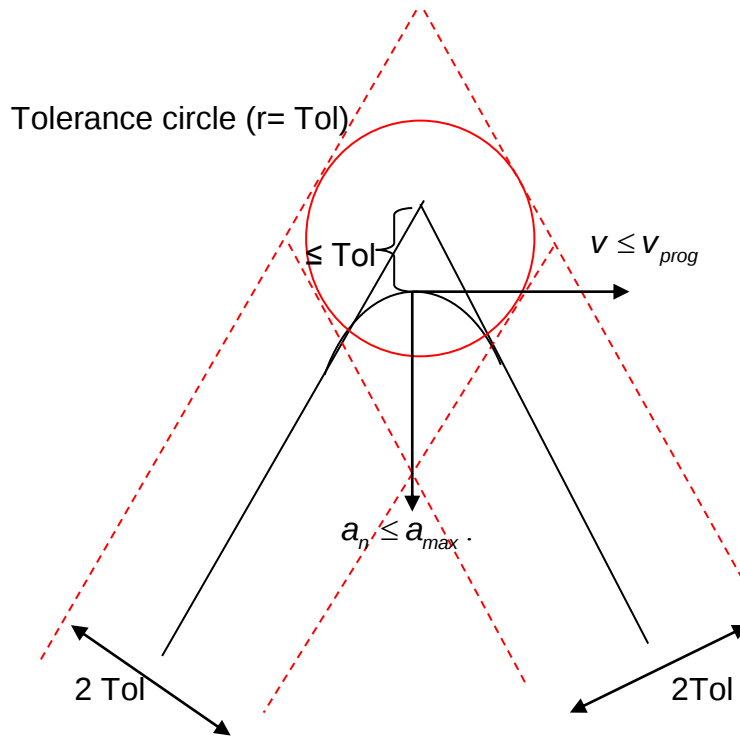


Figure 5. 6 Fitting an arc in the corner

5.4 Path Modification Using Average Filter

Instead of fitting arcs into corners, one can also modify the path using a running average filter with N samples, where N is related to the desired tolerance limit. The main advantage

of this strategy over the previous strategy is that it is implementable on any arbitrary nominal path. Also, this strategy is computationally more effective.

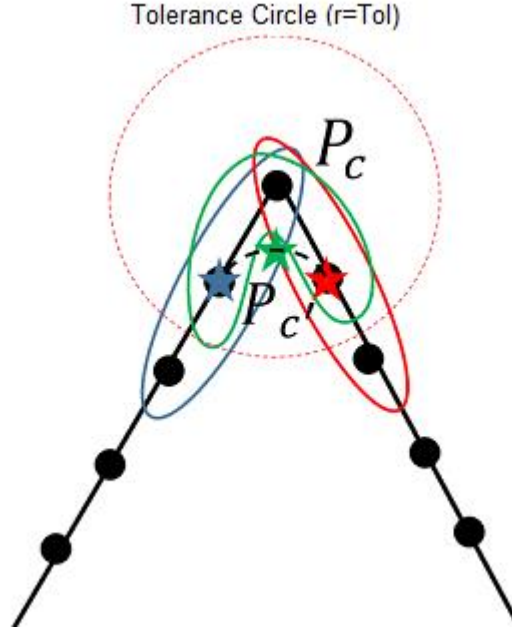


Figure 5. 7 Using average filter to modify a path

In Figure 5.7, running average filters over 3 samples are applied to the X and Y nominal positions respectively, resulting in the reference path indicated with star symbols on the dashed line in Figure 5.7. The blue, green, and red stars are averages of the nominal positions shown in the blue, green, and red ovals respectively.

Clearly, the distance between $P_{c'}$ and P_c should be smaller than the tolerance value, Tol . In order to calculate the distance, one can calculate the position difference in X and Y axes, and derive the geometric average of them:

$$Tol \geq \sqrt{(X_{c'} - X_c)^2 + (Y_{c'} - Y_c)^2}. \quad (5.7)$$

Maximum values of the distances in each axis are encountered in simultaneous reversals of both axes at maximum velocity. If the running average filter has an even number of samples, N , this leads to:

$$\frac{N\dot{X}_{max}\Delta T}{4} \geq (X_{c'} - X_c), \quad (5.8)$$

$$\frac{N\dot{Y}_{max}\Delta T}{4} \geq (Y_{c'} - Y_c). \quad (5.9)$$

Combining Equations (5.7-9) results in:

$$Tol^2 \geq \left(\frac{N\dot{X}_{max}\Delta T}{4}\right)^2 + \left(\frac{N\dot{Y}_{max}\Delta T}{4}\right)^2. \quad (5.10)$$

Assuming the same programmed velocities in the X and Y axes ($\dot{X}_{max} = \dot{Y}_{max} = v_{prog}$), one can derive the maximum length of the filter, N:

$$N_{max} = \frac{2\sqrt{2}Tol}{v_{prog}\Delta T}. \quad (5.11)$$

After creating the reference path, the feedrate modulation algorithm presented in Chapter 4 is utilized to select the reference velocities necessary to define the reference trajectory.

5.5 Experimental Setup

The experimental setup used to compare the two tolerance based trajectory generators in this chapter is identical to the setup used in Chapter 4. A PID controller with torque feedforward is used for the position control. Maximum velocities, maximum accelerations, and position loop bandwidth are selected according to Table 5.1. The interpolator is updated to include the algorithms described in sections 5.3 and 5.4.

Table 5. 1 Experimental setup

$\dot{X}_{max} = \dot{Y}_{max} = 0.12$	$\left[\frac{m}{s} \right]$	Maximum table speed
$\ddot{X}_{max} = \ddot{Y}_{max} = 5$	$\left[\frac{m}{s^2} \right]$	Maximum table acceleration
$\omega_c = 126$	$\left[\frac{rad}{s} \right]$	Controller bandwidth

5.6 Experimental Results

An orthogonal corner is chosen as the nominal path for the machine tool to follow. The linear distance before and after the corner is 2.5 mm. The programmed velocity is 0.12m/s.

To create a benchmark for the tolerance based interpolators, the nominal reference path with zero tolerance and the feedrate modulation algorithm described in Chapter 4 is chosen. Table 5.2 indicates that the benchmark scenario takes 82ms and results in a contouring error of $14\mu\text{m}$.

Since, both proposed strategies require predefined tolerance values to generate the reference paths, four tolerance values are selected for the experiments: $100\mu\text{m}$, $200\mu\text{m}$, $400\mu\text{m}$, and $800\mu\text{m}$. The nominal path, the modified reference path, and the actual path for different tolerance values are shown in Figures 5.8, and 5.9 for the arc fitting algorithm and the running average algorithm respectively.

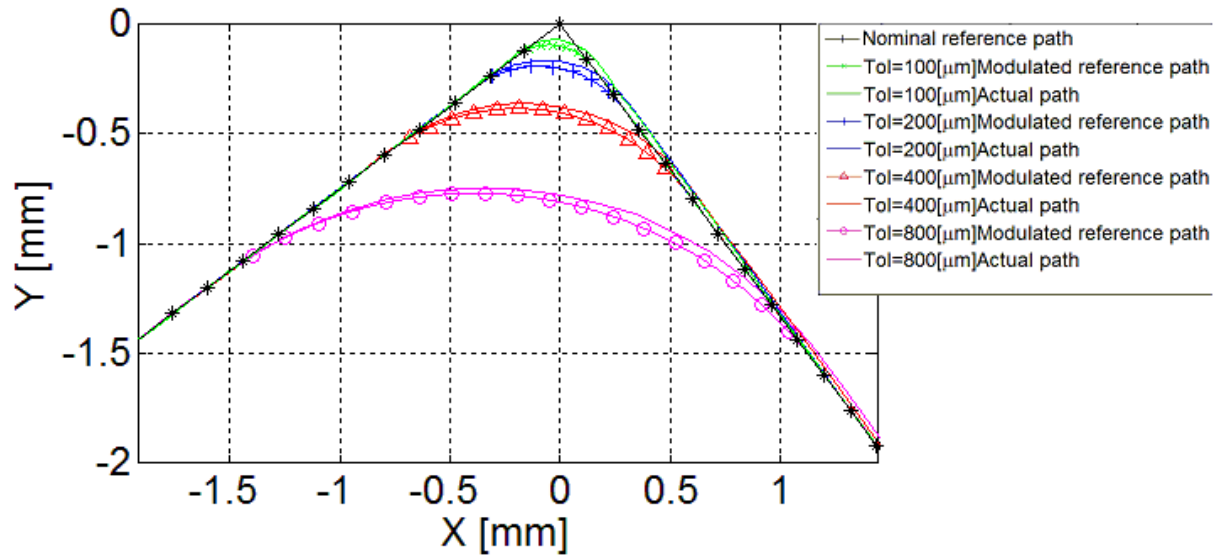


Figure 5. 8 Nominal path, modulated reference path, and actual path for the first strategy (fitting an arc in the corner)

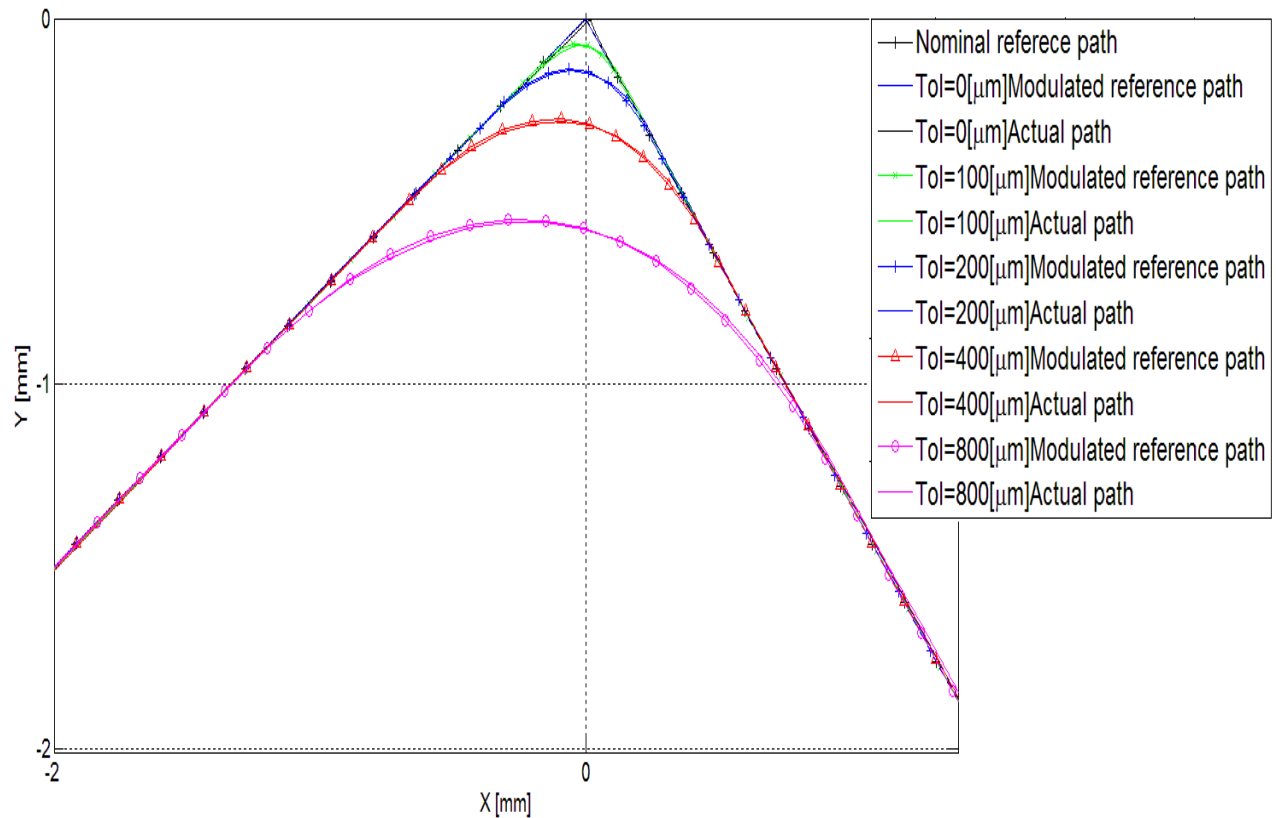


Figure 5. 9 The nominal path, the modified reference path, and the actual path for the second strategy (using running average filter)

Both proposed strategies successfully modify the nominal reference path and the machine tool is able to follow the created reference paths. The travel times and the resulting tolerances are recorded in Table 5.3. The travel time improvement is obtained by normalizing with the travel time obtained with the nominal reference trajectory and zero tolerance as listed in table 5.2.

Table 5. 2 Experimental results using nominal reference trajectory and the feedrate override method in Chapter 4

Tolerance values $[\mu m]$	Travel times using just feedrate override $[ms]$	Contouring Error values at the corner $[\mu m]$
0	81.7	13.36

Table 5. 3 Experimental results of two proposed algorithms

Tolerance values [μm]	Travel times Arc fitting strategy [ms]	Travel time improvement (%)	Contouring Error values at the corner [μm]	Travel times using running average filter [ms]	Travel time improvement (%)	Contouring Error values at the corner [μm]
100	70.6	1.40	77	68.5	4.33	80.33
200	67.3	6.01	173	66.1	7.68	152.81
400	61.1	14.66	373	61.8	13.69	294.77
800	55.7	22.21	773	55.7	22.21	575.29

Both new interpolation algorithms show a measureable improvement in travel time, while adhering to the prescribed tolerance levels. However, the arc fitting strategy leads to larger contour errors and longer travel times. It should be pointed out that the running average filter algorithm will lead to tolerances that are up to 1.4 times smaller than the desired tolerances. Only a biaxial reversal at maximum speed will lead to a path with maximum tolerance levels.

5.7 Discussions

This chapter proposed two new path/trajectory modification strategies. The first strategy is limited to corner tracking and fits arcs in corners of the nominal path using predefined tolerance values. This strategy has two main disadvantages: It is only applicable to corner tracking and it is computationally expensive. The second strategy proposed in this chapter uses running average filters based on predefined tolerance values. Both strategies are validated and compared experimentally with a nominal zero tolerance trajectory. The experiments indicate that both proposed strategies require less time than the nominal reference case with zero tolerance. However, the running average filter strategy outperforms the arc fitting strategy at smaller tolerance values, and it is more computationally efficient.

Chapter 6: Conclusions

The main focus of this study is on control and feedrate optimization strategies for high speed contouring of machine tools. Chapter 1 provides a general review of the challenges faced in high speed contouring. A detailed review on the published solution strategies for these challenges is provided in Chapters 2. In Chapter 3 conventional control strategies are benchmarked experimentally on a biaxial XY-table. In Chapter 4 a new real time feedrate optimization method is proposed that is able to incorporate velocity and acceleration constraints. In Chapter 5 two new reference path generation techniques are introduced that utilize allowable tolerance values to increase counteracting speed. The following provide a detailed summary of the results obtained in this thesis.

6.1 Experimental Comparison of Contouring Control Architectures

Chapter 3 provides a detailed study on conventional contouring control architectures. Studied architectures are divided into three major categories: cartesian servo controllers (CSC), contour error controllers, and feed forward controllers. For cartesian servo control (CSC), state feedback (SF), proportional derivative (PD), and proportional integral derivative (PID) feedback controllers are studied. As expected, PID control provides lowest static and dynamic errors, since it provides unity gain and constant group delay over a wider frequency range, as well as better disturbance rejection than SF and PD control. To improve contouring performance over CSC architectures, Cross Coupled Control (CCC) and Tangential normal Control (TC) have been suggested in the literature. CCC adds a control action normal to the path to conventional CSC. TC entirely replaces Cartesian control actions with normal and tangential control actions along the path. In our experiments, CCC and TC control do indeed provide better tracking performance than CSC, However, transient performance can be improved using feed forward strategies. In this thesis, Torque Feed Forward (TFF) and Zero Phase Error Tracking (ZPET) controllers are added to CSC and TC control for benchmarking with circles and corners. These experiments lead to three conclusions: 1.) ZPETC and TFF provide the same level of tracking performance, but TFF requires less computational effort; 2.) Both TFF and ZPETC require a PID feedback control law in order to guarantee zero

steady state tracking errors; 3.) The performance of a simple CSC+PID+TFF control law performs as well as the more complicated control architectures TC+PID+TFF, suggesting that CSC+PID+TFF should be used as the benchmark controller for future developments in high-speed contouring control.

6.2 A New Real Time Feedrate Optimization Algorithm for 2-D Cartesian Machine Tools

Chapter 4, proposes a new real time feedrate optimization method which is independent of the spline representation of the nominal path, and incorporates both velocity and acceleration constraints. The algorithm is built on the fact that a time optimal feedrate requires that either velocity or acceleration must meet its constraint values at each sample time. In this regard, for each pair of position increments, the algorithm adjusts the corresponding pair of sampling times by solving velocity and acceleration equations. The fact that the proposed algorithm is independent from the spline representation (since it simply adjusts sampling times for position increments), makes it more flexible than conventional algorithms in the literature that require particular spline representations of the reference trajectory. Another common problem of many traditional trajectory generation algorithms is that they perform a forward and a backward pass across the complete trajectory. This requires that the complete path be known before the trajectory can be generated. The algorithm shown in this thesis requires only a small look-ahead window. This allows for an easier implementation of the algorithm in practical machine tool control systems.

6.3 Reference Trajectories with Tolerance Bounds

Chapter 5 proposes two new trajectory generator algorithms using allowable tolerance values. The proposed algorithms attempt to adjust the tool path at critical points with large normal accelerations, which would require substantial velocity reductions in conventional trajectory generators. The first proposed algorithm is limited to corners. It fits arcs in corners within the tolerance bounds around the path. The second algorithm works for any trajectory type and not just corners. It utilizes a running average filter to smooth the tool path at critical

points. The length of the running average filter is determined by the allowable tolerances. An experimental study of the two new algorithms shows that the new trajectory generation algorithms can provide significant time savings when typical machining tolerances are incorporated into the tool trajectories. The experiments also show that the running average filter provides tool path that are both faster and easier to track than circles fitted into corners.

6.4 Future Works

This thesis benchmarks conventional contouring control strategies and suggests new trajectory generation algorithms. Possible future work based on the findings of this thesis includes:

- Studying more complex controllers: This study benchmarked conventional and well known controllers. It is suggested that CSC+PID+TFF control should be used as the benchmark to study the performance of more advanced controllers. The literature showed a large number of complex controllers for machine tools such as Model Predictive Control, Robust Control, or Fuzzy Control. Investigating their performance and comparing them to (PID+FF) is an interesting topic for future work.
- Generalized running average filter for trajectory generating: The running average filter used in Chapter 5 for reference path generation uses a filter with a constant length. This length is based on providing maximum tolerances at high speed reversals. However, this strategy will not take advantage of the full tolerance bounds at lower speeds. Thus, the length of the running average filter can be adjusted adaptively along the trajectory to provide faster contouring performance.
- Considering machining disturbances: The work shown in this thesis is based on the assumption that disturbances due to machining forces are small. This is accurate for machining processes such as laser machining where cutting forces are negligible. For conventional turning or milling, cutting forces can be substantial and might require real time modification of the tool feedrate. In the future, the algorithms proposed in this thesis should be combined with real time machining process identification

algorithms that would then adjust the feedrate in order to avoid tool breakage and extend tool life.

References

- [1] Kalpakjian, S., Schmid, S. R., & Kok, C. W. (2006). Manufacturing Engineering and technology. Pearson-Prentice Hall.
- [2] Kalpakjian, S., Schmid, S. R., & Kok, C. W. (2008). Manufacturing processes for engineering materials. Pearson-Prentice Hall.
- [3] Emami, M. M., & Arezoo, B. (2010). A look-ahead command generator with control over trajectory and chord error for NURBS curve with unknown arc length. *Computer-Aided Design*, 42(7), 625-632.
- [4] Nam, S. H., & Yang, M. Y. (2004). A study on a generalized parametric interpolator with real-time jerk-limited acceleration. *Computer-Aided Design*, 36(1), 27-36.
- [5] Lin, M. T., Tsai, M. S., & Yau, H. T. (2007). Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm. *International Journal of Machine Tools and Manufacture*, 47(15), 2246-2262.
- [6] Sun, Y., Jia, Z., Ren, F., & Guo, D. (2008). Adaptive feedrate scheduling for NC machining along curvilinear paths with improved kinematic and geometric properties. *The International Journal of Advanced Manufacturing Technology*, 36(1-2), 60-68.
- [7] Zhang, M., Yan, W., Yuan, C., Wang, D., & Gao, X. (2011). Curve fitting and optimal interpolation on CNC machines based on quadratic B-splines. *Science China Information Sciences*, 54(7), 1407-1418.
- [8] Altintas, Y., & Erkorkmaz, K. (2003). Feedrate optimization for spline interpolation in high speed machine tools. *CIRP Annals-Manufacturing Technology*, 52(1), 297-302.

- [9] Timar, S. D., Farouki, R. T., & Boyadjieff, C. L. (2007). TIME-OPTIMAL FEEDRATES ALONG CURVED PATHS FOR CARTESIAN CNC MACHINES WITH PRESCRIBED BOUNDS ON AXIS VELOCITIES AND ACCELERATIONS. *International Journal of Robotics and Automation*, 22(2).
- [10] Farouki, R. T., Manjunathaiah, J., & Yuan, G. F. (1999). G codes for the specification of Pythagorean-hodograph tool paths and associated feedrate functions on open-architecture CNC machines. *International Journal of Machine Tools and Manufacture*, 39(1), 123-142.
- [11] Timar, S. D., & Farouki, R. T. (2007). Time-optimal traversal of curved paths by Cartesian CNC machines under both constant and speed-dependent axis acceleration bounds. *Robotics and Computer-Integrated Manufacturing*, 23(5), 563-579.
- [12] Feng, J., Li, Y., Wang, Y., & Chen, M. (2010). Design of a real-time adaptive NURBS interpolator with axis acceleration limit. *The International Journal of Advanced Manufacturing Technology*, 48(1-4), 227-241.
- [13] Erkorkmaz, K., & Altintas, Y. (2001). High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation. *International Journal of machine tools and manufacture*, 41(9), 1323-1345.
- [14] Tarkiainen, M., & Shiller, Z. (1993, May). Time optimal motions of manipulators with actuator dynamics. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on* (pp. 725-730). IEEE.
- [15] Costantinescu, D., & Croft, E. A. (2000). Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *Journal of robotic systems*, 17(5), 233-249.
- [16] Dong, J., & 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), 379-390.

- [17] Seethaler, R. J., & Yellowley, I. (1996). The regulation of position error in contouring systems. *International Journal of machine tools and manufacture*, 36(6), 713-728.
- [18] Yang, L., & Yellowley, I. (2001). High-speed contouring using a novel dynamic interpolation mechanism. *International Journal of machine tools and manufacture*, 41(6), 773-794.
- [19] Oldknow, K. D., & Yellowley, I. (2003). Implementation and validation of 3-dimensional dynamic interpolation using an FPGA based controller. *International Journal of machine tools and manufacture*, 43(9), 937-945.
- [20] Rahaman, M., Seethaler, R., & Yellowley, I. (2015). A new approach to contour error control in high speed machining. *International Journal of Machine Tools and Manufacture*, 88, 42-50.
- [21] Ardekani, R., & Yellowley, I. (1996). The control of multiple constraints within an open architecture machine tool controller. *Journal of Manufacturing Science and Engineering*, 118(3), 388-393.
- [22] Seethaler, R. J., & Yellowley, I. (2000). Process control and dynamic process planning. *International Journal of Machine Tools and Manufacture*, 40(2), 239-257.
- [23] Yao, B., Hu, C., & Wang, Q. (2012). An orthogonal global task coordinate frame for contouring control of biaxial systems. *Mechatronics, IEEE/ASME Transactions on*, 17(4), 622-634.
- [24] Yang, J., & Li, Z. (2011). A novel contour error estimation for position loop-based cross-coupled control. *Mechatronics, IEEE/ASME Transactions on*, 16(4), 643-655.
- [25] Tomizuka, M. (1987). Zero phase error tracking algorithm for digital control. *Journal of Dynamic Systems, Measurement, and Control*, 109(1), 65-68.

- [26] Yamada, M., Funahashi, Y., & Fujiwara, S. I. (1996). Zero phase error tracking controller with a desired gain error and application.
- [27] Li, Y., & Xu, Q. (2010). Development and assessment of a novel decoupled XY parallel micropositioning platform. *Mechatronics, IEEE/ASME Transactions on*, 15(1), 125-135.
- [28] Koren, Y. (1980). Cross-coupled biaxial computer control for manufacturing systems. *Journal of Dynamic Systems, Measurement, and Control*, 102(4), 265-272.
- [29] Koren, Y., & Lo, C. C. (1991). Variable-gain cross-coupling controller for contouring. *CIRP Annals-Manufacturing Technology*, 40(1), 371-374.
- [30] Shih, Y. T., Chen, C. S., & Lee, A. C. (2002). A novel cross-coupling control design for bi-axis motion. *International Journal of machine tools and manufacture*, 42(14), 1539-1548.
- [31] Su, K. H., & Cheng, M. Y. (2008). Contouring accuracy improvement using cross-coupled control and position error compensator. *International Journal of Machine Tools and Manufacture*, 48(12), 1444-1453.
- [32] Srinivasan, K., & Kulkarni, P. K. (1990). Cross-coupled control of biaxial feed drive servomechanisms. *ASME Journal of Dynamic Systems, Measurement, and Control*, 112(4), 225-232.
- [33] Yeh, S. S., & Hsu, P. L. (1999). Theory and applications of the robust cross-coupled control design. *Journal of dynamic systems, measurement, and control*, 121(3), 524-530.
- [34] Shih, Y. T., Chen, C. S., & Lee, A. C. (2002). A novel cross-coupling control design for bi-axis motion. *International Journal of machine tools and manufacture*, 42(14), 1539-1548.
- [35] Tarn, Y. S., Chuang, H. Y., & Hsu, W. T. (1999). Intelligent cross-coupled fuzzy feedrate controller design for CNC machine tools based on genetic algorithms. *International Journal of machine tools and manufacture*, 39(10), 1673-1692.

- [36] Cheng, M. Y., Su, K. H., & Wang, S. F. (2009). Contour error reduction for free-form contour following tasks of biaxial motion control systems. *Robotics and Computer-Integrated Manufacturing*, 25(2), 323-333.
- [37] Chen, C. S., Fan, Y. H., & Tseng, S. P. (2006). Position command shaping control in a retrofitted milling machine. *International Journal of Machine Tools and Manufacture*, 46(3), 293-303.
- [38] Barton, K., & Alleyne, A. (2006). Cross-coupled iterative learning control: Design and implementation. In *Proceedings of IMECE*.
- [39] Barton, K. L., & Alleyne, A. G. (2008). A cross-coupled iterative learning control design for precision motion control. *Control Systems Technology, IEEE Transactions on*, 16(6), 1218-1231.
- [40] Conway, J. R., Ernesto, C. A., Farouki, R. T., & Zhang, M. (2012). Performance analysis of cross-coupled controllers for CNC machines based upon precise real-time contour error measurement. *International Journal of Machine Tools and Manufacture*, 52(1), 30-39.
- [41] Chiu, G. C., & Yao, B. (1997, June). Adaptive robust contour tracking of machine tool feed drive systems-a task coordinate frame approach. In *American Control Conference, 1997. Proceedings of the 1997* (Vol. 5, pp. 2731-2735). IEEE.
- [42] Chiu, G. C., & Tomizuka, M. (2001). Contouring control of machine tool feed drive systems: a task coordinate frame approach. *Control Systems Technology, IEEE Transactions on*, 9(1), 130-139.
- [43] Lo, C. C., & Chung, C. Y. (1999). Tangential-contouring controller for biaxial motion control. *Journal of dynamic systems, measurement, and control*, 121(1), 126-129.

- [44] Zhang, D., Lou, Y., & Li, Z. (2006, October). Geometric contouring control on the smooth surface. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (pp. 4496-4501). IEEE.
- [45] Chen, C. L., & Lin, K. C. (2008). Observer-based contouring controller design of a biaxial stage system subject to friction. *Control Systems Technology, IEEE Transactions on*, 16(2), 322-329.
- [46] Hu, C., Yao, B., & Wang, Q. (2010). Coordinated adaptive robust contouring control of an industrial biaxial precision gantry with cogging force compensations. *Industrial Electronics, IEEE Transactions on*, 57(5), 1746-1754.
- [47] Lou, Y., Li, Z., Zhong, Y., Li, J., & Li, Z. (2011). Dynamics and contouring control of a 3-DOF parallel kinematics machine. *Mechatronics*, 21(1), 215-226.
- [48] Hu, C., Yao, B., & Wang, Q. (2011). Global task coordinate frame-based contouring control of linear-motor-driven biaxial systems with accurate parameter estimations. *Industrial Electronics, IEEE Transactions on*, 58(11), 5195-5205.
- [49] Yao, B., Hu, C., & Wang, Q. (2012). An orthogonal global task coordinate frame for contouring control of biaxial systems. *Mechatronics, IEEE/ASME Transactions on*, 17(4), 622-634.
- [50] Lou, Y., Meng, H., Yang, J., Li, Z., Gao, J., & Chen, X. (2014). Task polar coordinate frame-based contouring control of biaxial systems. *Industrial Electronics, IEEE Transactions on*, 61(7), 3490-3501.
- [51] Chen, S. L., Liu, H. L., & Ting, S. C. (2002). Contouring control of biaxial systems based on polar coordinates. *Mechatronics, IEEE/ASME Transactions on*, 7(3), 329-345.

- [52] Bieterman, M. B., & Sandstrom, D. R. (2002, January). A curvilinear tool-path method for pocket machining. In *ASME 2002 International Mechanical Engineering Congress and Exposition* (pp. 149-158). American Society of Mechanical Engineers.
- [53] Dong, J., Ferreira, P. M., & Stori, J. A. (2007). Feed-rate optimization with jerk constraints for generating minimum-time trajectories. *International Journal of Machine Tools and Manufacture*, 47(12), 1941-1955.
- [54] Dong, J., & Stori, J. A. (2007). Optimal feed-rate scheduling for high-speed contouring. *Journal of manufacturing science and engineering*, 129(1), 63-76.
- [55] Seethaler, R. J., & Yellowley, I. (1996). The regulation of position error in contouring systems. *International Journal of machine tools and manufacture*, 36(6), 713-728.
- [56] Bayo, E. (1988, April). Computed torque for the position control of open-chain flexible robots. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on* (pp. 316-321). IEEE.
- [57] Onaga, E. M., Casler Jr, R. J., Penkar, R. C., Lancraft, R. E., & Sha, C. (1990). *U.S. Patent No. 4,925,312*. Washington, DC: U.S. Patent and Trademark Office.
- [58] Xi, X. C., Zhao, W. S., & Poo, A. N. (2015). Improving CNC contouring accuracy by robust digital integral sliding mode control. *International Journal of Machine Tools and Manufacture*, 88, 51-61.
- [59] Zhao, G., Shen, Y., & Wang, Y. (2013). Fuzzy PID position control approach in computer numerical control machine tool. *Journal of Computers*, 8(3), 622-629.
- [60] Adnan, M. M., Sarkheyli, A., Zain, A. M., & Haron, H. (2013). Fuzzy logic for modeling machining process: A review. *Artificial Intelligence Review*, 1-35.

- [61] Rober, S. J., Shin, Y. C., & Nwokah, O. D. I. (1997). A digital robust controller for cutting force control in the end milling process. *Journal of dynamic systems, measurement, and control*, 119(2), 146-152.
- [62] Rahaman, M., Seethaler, R., & Yellowley, I. (2015). A new approach to contour error control in high speed machining. *International Journal of Machine Tools and Manufacture*, 88, 42-50.
- [63] Altintas, Y., Erkorkmaz, K., & Zhu, W. H. (2000). Sliding mode controller design for high speed feed drives. *CIRP Annals-Manufacturing Technology*, 49(1), 265-270.
- [64] Jee, S., & Koren, Y. (2004). Adaptive fuzzy logic controller for feed drives of a CNC machine tool. *Mechatronics*, 14(3), 299-326.
- [65] Yao, B. (1997, December). High performance adaptive robust control of nonlinear systems: a general framework and new schemes. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on* (Vol. 3, pp. 2489-2494). IEEE.
- [66] Yao, B., & Tomizuka, M. (1996). Smooth robust adaptive sliding mode control of manipulators with guaranteed transient performance. *Journal of dynamic systems, measurement, and control*, 118(4), 764-775.
- [67] Yao, B., & Tomizuka, M. (1997). Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form. *Automatica*, 33(5), 893-900.
- [68] Hao, Z., & Limei, W. (2014, May). Model predictive contouring control for direct drive XY table. In *Control and Decision Conference (2014 CCDC), The 26th Chinese* (pp. 3430-3433). IEEE.
- [69] Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Sokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789-814.

[70] Chang, S. H., & Herrin, G. D. (1991). Surface profile error evaluation procedures using a coordinate measuring machine.

[71] Erkorkmaz, K., & Altintas, Y. (2001). High speed CNC system design. Part II: modeling and identification of feed drives. *International Journal of Machine Tools and Manufacture*, 41(10), 1487-1509.

[72] Bosetti, P., & Bertolazzi, E. (2014). Feed-rate and trajectory optimization for CNC machine tools. *Robotics and Computer-Integrated Manufacturing*, 30(6), 667-677.

[73] Ernesto, C. A., & Farouki, R. T. (2012). High-speed cornering by CNC machines under prescribed bounds on axis accelerations and toolpath contour error. *The International Journal of Advanced Manufacturing Technology*, 58(1-4), 327-338.