TRAJECTORY GENERATION, CONTROL, AND GEOMETRIC ERROR COMPENSATION FOR A 9-AXIS MICROMACHINING CENTER

by

Alexander Yuen

B.A.Sc., The University of British Columbia, 2011
M.A.Sc., The University of British Columbia, 2013

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

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

July 2018

The following individuals certify that they have read and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, a thesis/dissertation entitled:

TRAJECTORY GENERATION, CONTROL, AND GEOMETRIC ERROR COMPENSATION FOR A 9-AXIS MICROMACHINING CENTER

submitted by Alexander Yuen in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering

Supervisory Committee Members:

Yusuf Altintas

Supervisor

Tim Salcudean

Supervisory Committee Member

Xiaodong Lu

Supervisory Committee Member

Elizabeth Croft

Supervisory Committee Member

Abstract

This thesis presents a trajectory generation algorithm, a control strategy, and a geometric error compensation methodology for a novel 9-axis micromachining center which combines a 3-axis micromill with a 6 degree of freedom magnetically levitated rotary table. The proposed trajectory generation algorithm resolves redundant degrees of freedom by numerically solving for axes positions from desired tool positions and orientations. Differential axes positions are found while ensuring the stroke limits of the drives are respected and singularities are avoided. The differential solution is numerically integrated to obtain the axes positions with respect to displacement. The axes commands are then scheduled in time, while respecting the velocity, acceleration, and jerk limits of each of the drives, and traversing the toolpath as fast as possible. The experiments showed trajectories that resolved redundancies, avoided singularities, and respected all physical limits of the drives.

A control strategy which combines the capabilities of the micromill and the rotary table is introduced. A sliding mode controller with a LuGre friction compensator is designed to control the position of the micromill, based on identified physical parameters. A lead-lag position controller with an integrator and a notch filter is designed to control the rotary table. Since the translational axes of the micromill and rotary table are in parallel, the tracking error of the micromill is sent as a reference command to the rotary table, compensating the tracking errors of the micromill with the higher bandwidth of the rotary table. In experiments, the dual stage control law improved tracking error over the micromill alone.

The geometric errors of the 3-axis micromill is compensated by using the precision motion of the 6 degree of freedom rotary table. The geometric errors of the 3-axis micromill are measured with a laser interferometer, fit to quintic polynomials, and incorporated into the forward kinematic model. The tooltip deviation is found by subtracting the ideal tooltip position from the tooltip position affected by geometric errors. Rotary table commands, from all 6 axes, that compensate for these deviations are found using a gradient descent algorithm. Experiments showed reductions in end effector deviations.

Lay Summary

Technological advances in industries such as the electronics and biomedical field has driven the demand for manufactured parts with high precision features, typically in the order of one thousandths of a millimeter. Subsequently there is a demand for manufacturing processes and machine tools capable of generating high precision features. A hybrid 9-axis machine tool, which combines a 3-axis micromilling machine and a 6 degree of freedom magnetically levitated rotary table has been developed for this purpose.

This thesis presents algorithms to plan the motion, control the precision, and control the accuracy for this 9-axis machine tool. The motion is planned to ensure all 9-axis are used as efficiently as possible given a predefined path. Furthermore the machine is controlled so the precision and accuracy of the 3-axis micromill is improved with the rotary table. The developed algorithms can be used with similar machines to manufacture high precision parts.

Preface

Chapter 3 and 4. A version of this material has been published in the CIRP Annals [Yuen A., Altintas Y. Trajectory generation and control of a 9-axis CNC micromachining center. CIRP Annals. doi: 10.1016/j.cirp.2016.04.098] and IEEE/ASME Transactions on Mechatronics [Yuen A., Altintas Y. Constrained Trajectory Generation and Control for a 9-Axis Micromachining Center With Four Redundant Axes. IEEE/ASME Transactions on Mechatronics. doi: 10.1109/TMECH.2017.2771260]. I was the lead investigator, responsible for the major development of the algorithms, experiment design, data collection and analysis, as well as manuscript composition. Altintas Y. was the supervisory author on this project and was involved throughout the project in concept formation and manuscript composition

Chapter 5. A version of this material has been published in the ASME Journal of Manufacturing Science and Engineering [Yuen A., Altintas Y. Geometric error compensation with a 6 degree of freedom rotary magnetic actuator. ASME Journal of Manufacturing Science and Engineering. doi:10.1115/1.4040938]. I was the lead investigator, responsible for the major development of the algorithms, experiment design, data collection and analysis, as well as manuscript composition. Altintas Y. was the supervisory author on this project and was involved throughout the project in concept formation and manuscript composition

Chapter 6. The results found in this chapter were previously included in the above three mentioned works. Modifications include a more detailed analysis of the results. All figures and tables found in this thesis are used with permission from applicable sources.

Table of Contents

Appendices

List of Tables

List of Figures

Nomenclature

C_c	Micromill controller
C_f	Rotary table controller
E_q	Error matrix for the q axis
F_c	Averaged Coulomb friction for LuGre friction model
F_s	Averaged static friction for LuGre friction model
$F_\text{coul}^{+/-}$	Coulomb friction with forward/reverse velocity
$F_\text{stat}^{+/-}$	Static friction with forward/reverse velocity
G_c	Plant dynamics of micromill feed drive
G_f	Plant dynamics of rotary table
$G_{c,cl}$	Closed loop transfer function of micromill feed drive
$G_{f,cl}$	Closed loop transfer function of rotary table
H	Cost function constraining the stroke limits of all 9 drives in redundancy resolution
J	Jacobian matrix of the forward kinematic model
$J^\dagger$	Moore-Penrose inverse of the Jacobian matrix J
$J^{c\dagger}$	Moore-Penrose inverse of the Jacobian matrix J^c
J^c	Jacobian matrix of the tooltip positions as a function of rotary table positions

$J_{x,y,z,a,b,c}$ Inertia of x, y, z, a, b, and c axis of rotary table

K_a Amplifier gain

K_f Gain of lead-lag controller

K_i Gain of integrator

K_t Motor gain

N_a Notch filter to remove resonance

N_b Notch filter to remove anti-resonance

$N_{i,p}$ B-spline basis function for feed profile with degree p

O_i Orientation of the tool about the X-axis of the workpiece coordinate frame

O_j Orientation of the tool about the Y-axis of the workpiece coordinate frame

O_k Orientation of the tool about the Z-axis of the workpiece coordinate frame

P_i Control points for feed profile

P_x X position of tooltip in the work piece coordinate frame

P_y Y position of tooltip in the work piece coordinate frame

P_z Z position of tooltip in the work piece coordinate frame

$P_{x,a}$ X position of tooltip in the work piece coordinate frame considering geometric errors

$P_{x,i}$ Ideal X position of tooltip in the work piece coordinate frame

$P_{y,a}$ Y position of tooltip in the work piece coordinate frame considering geometric errors

$P_{y,i}$	Ideal Y position of tooltip in the work piece coordinate frame
$P_{z,a}$	Z position of tooltip in the work piece coordinate frame considering geometric errors
$P_{z,i}$	Ideal Z position of tooltip in the work piece coordinate frame
S	B-spline basis function with
$S_{c,cl}$	Sensitivity transfer function of micromill feed drive
T	Constant to control frequency where phase is added with lead-lag controller
T_s	Sample time of servo system
V_s	Input signal
$\Delta\mathbf{P}$	Error between the ideal tooltip position and tooltip position considering geometric errors
$\Omega_1^{+/-}$	Input signal
$\Omega_2^{+/-}$	Input signal
α	Constant to control how much phase to add with lead-lag controller
α	Time index
$\dddot{\mathbf{q}}$	Axis jerks
$\dddot{\mathbf{q}}_{\text{max}}$	Axis jerk constraints
$\ddot{\mathbf{q}}$	Axis accelerations
$\ddot{\mathbf{q}}_{\text{max}}$	Axis acceleration constraints

$\delta_x(q)$	Error in the x-direction for the q axis
$\delta_y(q)$	Error in the y-direction for the q axis
$\delta_z(q)$	Error in the z-direction for the q axis
$\dot{\mathbf{q}}$	Axis velocities
$\dot{\mathbf{q}}_{\text{max}}$	Axis velocity constraints
$\dot{s}(s)$	Feed profile defined as a function of displacement
ϵ	Measured general geometric error
$\hat{\epsilon}\left(q\right)$	Estimated general geometric error function
λ	Bandwidth of sliding mode controller
$\mathbf{q}$	Vector of positions of all axes
$\mathbf{q}_f$	Vector of positions of rotary table axes
$\mathbf{q}_f$	Vector of rotary table positions
μ	Singularity avoidance constant
μ_0	Singularity avoidance constant limit
ω_a	Resonant frequency
ω_b	Anti-resonant frequency
ω_c	Bandwidth of micromill feed drive
ω_f	Bandwidth of rotary table

ω_m	Design cross-over frequency of negative loop transmission of rotary table
ρ	Disturbance adaption gain
σ	Sliding surface for sliding mode controller
σ_0	Stiffness of microstructures for LuGre friction model
σ_1	Damping of microstructures for LuGre friction model
c_q	$\cos(q)$
s_q	$\sin(q)$
$\varepsilon_x(q)$	Error about the x-direction for the q axis
$\varepsilon_y(q)$	Error about the y-direction for the q axis
$\varepsilon_z(q)$	Error about the z-direction for the q axis
ϑ	Spline parameter of feed profile B-spline function
$\zeta_{a,1/2}$	Damping ratios to control size and width of Notch filter N_a
$\zeta_{b,1/2}$	Damping ratios to control size and width of Notch filter N_b
a_f	Position of the a-axis of the rotary table
b	Feed drive viscous friction
b_f	Position of the b-axis of the rotary table
c_f	Position of the c-axis of the rotary table
k	Time index

$k_{1,2,3,4}$	Runge Kutta numerical integration constants
m	Feed drive mass
n	Displacement index
q	General axis position
s	Tangential displacement along toolpath
u	Input signal
w	Manipulability of 9-axis micromachining center
w_0	Threshold of manipulability to begin singularity avoidance
x_c	Position of the x-axis of the micromill
x_f	Position of the x-axis of the rotary table
y_c	Position of the y-axis of the micromill
y_f	Position of the y-axis of the rotary table
z_c	Position of the z-axis of the micromill
z_f	Position of the z-axis of the rotary table

Acknowledgments

I am most grateful to my research supervisor, Dr. Yusuf Altintas for his support and guidance over the course of the last 7 years. The insight and opportunities into the world of manufacturing engineering provided by Dr. Altintas were second to none and through my experience working with Dr. Altintas, I have matured as an engineer and an academic researcher in a way I never thought possible.

I would also like to thank all members of the Manufacturing Automation Laboratory family. Whether it was a current member helping overcome research challenges or an alumni providing guidance, members of the MAL family never hesitated to offer a helping hand, and for that, I am forever grateful.

Finally, I would like to thank my mother, father, family, and friends. Without their moral support and sacrifices none of this would have been possible. I dedicate this thesis to my grandfather, Dominic Leung, who lives in my heart.

This research was sponsored by the Natural Science and Engineering Research Council of Canadian Network for Research and Innovation in Machining Technology (NSERC CANRIMT).

Chapter 1

Introduction

In recent years there has been a decrease in the average size and increase in the complexity of manufactured parts due to technological progress in industries such as the biomedical and electronics industry. At present, it is common to see manufactured parts with features in the micrometer scale. Though multiple manufacturing processes with micron level precision do exist, few are capable of producing parts with freeform features with precision. Wire electrical discharge machining (EDM) and photolithography are capable of very high precisions but can only be used to generate 2-D and 2.5-D parts, respectively. Though micro injection molding and micro forming are capable of producing high precision free form parts, a higher precision manufacturing process must be used to produce the tooling. Given the limitations of the above mentioned manufacturing processes, multi-axis micromachining is one of the few viable means for producing freeform parts with micron level precision. This work contributes a development of novel multi-axis micromachine tool, which combines a traditional 3-axis micromill with a 6 degrees of freedom (DOF) magnetically actuated rotary table shown in Fig. 1.1.

The 6DOF rotary table was originally developed by the Precision Mechatronics Laboratory at the University of British Columbia and has been presented in [1] and [2]. The rotary table consists of a Halbach magnetic array as the mover and a printed circuit board (PCB) as a stator. Actuation is achieved by passing current through the copper traces in the PCB, which creates an opposing magnetic field to the Halbach magnetic arrays on the mover. This configuration allows for a tetherless mover and a low form factor stator. The rotary table has a full 6 degrees

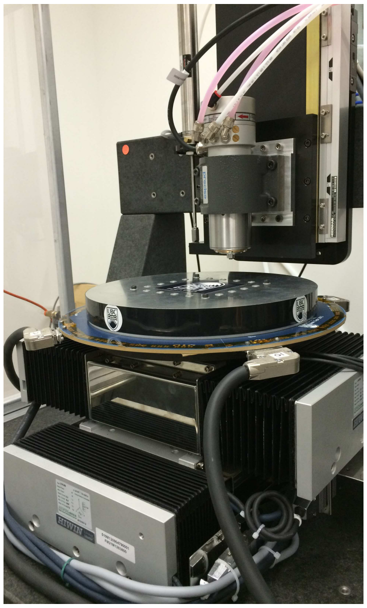


Figure 1.1: UBC MAL's hybrid 9-axis machine

of freedom with approximately 100 micron stroke limit in the x, y, and z directions, 0.5 degrees tilt limit about the x and y axis, and unlimited rotation when rotating about the z-axis.

By combining a lighter, and subsequently faster, actuator with the existing 3-axis micromill, it is possible to increase the precision and accuracy of the overall system. However, with this new configuration, the addition of multiple redundant axes introduce new research challenges.

Typically, a 5-axis machine is capable of moving a cutting tool to a predefined cutter location defined by 3 positions and 3 orientations. Since a 5-axis machine tool has the necessary degrees of freedom, analytic inverse kinematic solutions exist, allowing for unique mappings of machine tool axes positions to cutter locations. Conversely, the hybrid machine tool presented in this work has 4 more axes than necessary. As a result, a range of machine tool axes positions can correspond to a predefined cutter location. With respect to trajectory generation, the main challenge is selecting axes configurations that resolve the redundancies. Furthermore, the trajectory must also avoid singularities and be scheduled in a way to avoid the physical limits of the drives including the stroke, velocity, acceleration, and jerk limits. This thesis presents a novel trajectory generation algorithm to avoid singularities while respecting the limits of all axes.

With the trajectory defined, it is necessary to ensure the machine tool follows these trajectories as close as possible. This is typically done by applying high bandwidth position controllers to each axis of the machine tool. Though sophisticated modeling and control techniques can be used to increase the bandwidth to be as high as possible, the achievable bandwidth is limited by the stability of the closed loop controller of the axis. Likewise, in the presented configuration, the 3-axis micromill is controlled with a high performance non-linear controller, but is limited due to stability issues. Due to its comparatively lighter mass and lack of mechanical contact, the rotary table is able to achieve a bandwidth that is an order of magnitude higher than the 3-axis micromill. Since the translational axes of the rotary table and micromill run in parallel, it is possible to use the higher bandwidth of the rotary table to compensate the tracking errors contributed by the low bandwidth of the micromill. The thesis presents a dual stage feed drive control law that combines the strengths of the micromill and rotary table.

In addition to increasing the tracking precision of the machine tool, it is possible to improve the volumetric accuracy of the 3-axis micromill using the 6DOF rotary table. Though a feed drive should be constrained to only move in the direction of actuation, due to errors in assembly it is possible for the feed drive to deviate slightly in the orthogonal directions, in the translational and rotational sense. Furthermore, these errors are typically not detected since the feed drives have feedback on the direction of actuation. Though the deviations are small for an individual axis, when multiple axes in series have assembly errors, it can correspond to an end effector deviation that violates the tolerance of the parts to be machined. In this thesis, the rotary table is used to compensate for the affect of these position dependent geometric errors. The errors are measured with a laser interforemeter and mapped to end effector deviations. The trajectory of the 6DOF rotary table is then modified with a novel algorithm to compensate for these geometric errors.

Henceforth, the thesis is structured as follows: Chapter 2 discusses previous works reported in literature specifically with regards to trajectory generation for redundant actuators, dual stage feed drive control, and geometric error compensation. In Chapter 3 a novel trajectory generation algorithm is presented. The trajectory generation method resolves the redundant degrees of freedom while ensuring that the generated motion commands are jerk continuous, time optimal, and do not violate the stroke, velocity, acceleration, and jerk limits of the drive. The modeling, identification and development of the control laws used to control the micromill, rotary table, and the combined efforts of the micromill and rotary table are presented in Chapter 4. The higher bandwidth of the rotary table is used to compensate for the tracking errors from the lower bandwidth of the micromill. Chapter 5 provides an application of the rotary table, where the geometric errors of the micromill are first modeled through experimental measurements and compensated by modifying the trajectory of the rotary table. Chapter 6 presents simulation and experimental results and Chapter 7 concludes the thesis and suggests future reserach.

Chapter 2

Literature Review

2.1 Overview

The configuration in the work presented is a 9-axis micromachining center. Due to its unique configuration, new trajectory generation algorithms, control laws, and accuracy enhancement techniques must be developed. With regards to trajectory generation, since there are 4 more degrees of freedom than necessary the main challenge is in resolving these redundant degrees of freedom. In the presented configuration, the translational axes of the micromill are parallel with the translational axes of the rotary table. As a result, control laws must be developed that exploit the strengths of the different actuators, which include the long stroke of the micromill axis and the high bandwidth of the rotary table axis. Finally, the additional degrees of freedom can also be used to compensate for geometric errors coming from the 3-axis micromill. As a result, this literature survey evaluates existing work in trajectory generation of redundant mechanisms, control law design for machine tools and dual-stage servomechanisms, and the modeling, identification, and compensation of geometric errors.

2.2 Trajectory Generation for Redundant Manipulators and Machine tools

Typically, a cutting tool only needs 5 degrees of freedom to define its position and orientation in space. Furthermore, it is the end effector position and orientation that is typically specified since it is the element of the overall system that is interacting with its environment. As a result, one of the main objectives of trajectory generation, in the context of redundant manipulators, is resolving the redundant joints or axes so the end effector position and orientation corresponds to the desired position and orientation. In machine tools, there are typically 5 or less axes, and as a result an analytic solution for this problem is typically available in the form of inverse kinematics [3–5]. However, robotic devices typically have more joints than necessary as seen in various types of robotics arms. As a result, the same analytic inverse kinematic solutions can not be used for these under constrained configurations. In literature, the earliest work for the trajectory generation of redundant manipulators come from the field of robotics. This problem was initially solved by Whitney [6], where given a redundant manipulator and specified end effector velocities, the velocities of the joints are solved using the Moore-Penrose inverse of the robot's Jacobian matrix. Since the Moore-Penrose inverse only considers the least norm solution of the redundancy resolution problem, an infinite range of solution still exists. Leigios extended this work in [7], where the nullspace of the Jacobian matrix is used to minimize some cost function. By considering the nullspace of the Jacobian matrix, the desired end effector position is not affected, but the joint configuration is selected so the predefined cost function is minimized at each step.

In Leigios' original work, the cost function is specified so the solution selects a joint configuration that always respects the stroke limits of the joints. Numerous authors have extended Leigios' original idea and used the cost function to achieve secondary goals such as the optimization of joint torques [8], minimization of energy consumption [9], or avoidance of singularities [10]. In these solutions, the differential solution minimizes the specified cost function to a local optimal minimum at each solution step. Nakamura and Hanafusa [11] extended these types of solution to consider global optimality of the cost function. In [11], Nakamura and Hanafusa used the example of minimizing manipulability over the whole toolpath. It should be noted that due to the recursive nature of these globally optimal solution, the computation time was orders of magnitude higher than with the locally optimal solution. It should be noted that the redundancy is resolved at the differential level, and as a result, there is a need to numerically integrate the solution in order to obtain actual motion commands. In [12], the error differences between the various numerical integration techniques were investigated. It was found that the 4th Order Runge Kutta Method produced the best results. In order to improve the numerical integration result, a closed loop corrective element is added in [13]. This way, the numerical drift inherent in the use of numerical integration techniques is corrected at each step.

Though the above methods are able to generate trajectories for redundant actuators, the resultant trajectories are not time optimal trajectories that consider the joint velocity, acceleration, and jerk limits. It should be noted that literature exists from the robotics field for optimal trajectory generation [14] [15], but these works do not address kinematically redundant configurations. Since robotic applications do not require the same productivity demands as manufacturing applications, the need for time optimality is not as important and it is sufficient for a robot to operate under its maximum capacity. In contrast, trajectory generation algorithms for machine tools must be as time optimal as possible, at least acceleration continuous, and respect the physical limits of all the drives. As a result, trajectory generation techniques in machine tool literature are focused in fulfilling these criteria in different ways. In [16], Dong et al. proposed a solution to find a time optimal jerk limited trajectory with a bi-directional scan to optimize the trajectory only for jerk and a secondary acceleration-continuation algorithm to optimize the trajectory for acceleration. In [17] a solution is proposed where a tangential displacement is selected at each servo time step in a way that saturates one constraint at all times. The solutions proposed in [16] and [17] are only acceleration continuous. In [18] and [5], similar constraints on velocity, acceleration and jerk are used. However, the tangential tool path is jerk continuous. Due to this added complexity, non-linear optimization techniques were used, where the feedrate is optimized while respecting the constraints by modifying segment durations in [18] and the control points of a spline that defines the feedrate profile in [5].

Based on this literature survey it can be seen that there exists a gap in literature. At present, techniques have not been developed where time optimal, smooth, and constrained trajectories can be generated for redundant configurations.

2.3 Control laws for Feed drives and Dual Stage Actuators

Due to the need for high precision positioning, a rich body of literature exist in the field of single axis feed drive control for machine tools beyond the standard industrial Proportional-Integral-Derivative (PID) controllers. Erkorkmaz et al. presented a sophisticated feed drive controller in [19] which used a combination of zero phase error tracking control [20], pole placement, Kalman filter, and feed forward friction compensation. Altintas et al. [21] developed a high bandwidth, robust sliding mode controller with disturbance compensation showing similar results as those presented in [19] but with a simpler control structure. Okuwdire extended the work in [22] and included the flexible modes of the ball screw in the design of the sliding mode controller. A switching gain scheduling controller is proposed in [23], which accommodates for position dependent dynamics and variations in mass. Hosseinabadi and Altintas [24] used an active damping network to damp the structural modes of the machine tool to increase the bandwidth of the sliding mode controller originally proposed in [19]. In [25], Kamalzadeh and Erkorkmaz compensated for the axial vibrations of the ballscrew drive by including the first axial mode into the sliding mode control law and showed superior performance over the use of a notch filter. Kamalzadeh and Erkorkmaz combined multiple control techniques in [26] and [27], where excitation of torsional modes are avoided with notch filters, control of rigid body dynamics is achieved with an adaptive sliding mode controller, and non-linear friction and torque ripple are modeled and compensated in a feedforward fashion. It should be noted that the above works represent only a small subset of literature available in the field of feed drive controls.

In contrast, due to its relative uniqueness, particularly in the field of manufacturing, there is less work for dual stage feed drives. Preliminary work was done in [28] where the tracking error of the coarse stage is compensated with the fine stage. This work was extended in [29], where the estimated contouring error of the coarse stage is sent to the fine stage instead of the tracking error. In these works, a simple PID or PPI controller are used for the feedback control. Alfizy et al. [30] proposed a dual stage feed drive with a magnetic coarse actuation stage and a piezoelectric driven flexture fine actuation stage. In this work, the coarse stage is controlled with a simple PID controller and the fine stage compensates for the tracking error of the coarse stage. In [31], Choi et al. proposed a dual servo stage for UV lithography. Unlike the configurations presented in [28–30], where the position of the fine actuator is measured relative to the coarse actuator, the position of the fine actuator is monitored with laser interferometer relative to the machine tool's base coordinate frame. Based on the ability to measure the fine actuator with a higher precision sensor relative to the base coordinate frame, the control relationship between the coarse and fine actuator is switched, and the coarse actuator follows the tracking error of the fine actuator. Each axis is then controlled with a PID controller. It should be noted that the use of laser interferometer to measure positions in a CNC milling machine tool would be impractical due to much larger actuating range resulting in difficulties in mounting mirrors. In contrast to the multi-axis works presented above, in 2004, Kim et al. [32] developed a dual stage actuator and robust controller for camshaft turning. Given the repetitive nature of the input signal, the controller is designed to converge to an optimal design after a certain amount of time.

Beyond the field of multi-axis manufacturing machine tools, dual stage configurations are more commonly found in 1 DOF read heads of hard disk drives. In these setups, a voice coil motor drives the coarse movement and a piezoelectric actuator drives the fine movement. As a result, the literature for the control of hard disk drive read heads is prevalent with examples of dual stage control. It should be noted that machine tools and read heads of hard disk drives have different control objectives and configurations. Unlike machine tools, where the axes typically follow a smooth varying trajectory, read heads are typically given step commands to different tracks located on the disk. As a result the objective is to react as fast as possible to a step command, as opposed to following a smooth trajectory as close as possible. Furthermore, unlike machine tools, where each individual axis has their own respective feedback sensor, the feedback sensor detects the position of the read head, the positional sum of the voice coil motor and piezoelectric actuator, allowing the closed loop control algorithm to control the actual position of the end effector. As a result, the control laws developed address the control objectives and dynamic analysis in a different way. In [33] the authors implemented a linear controller for the dual-stage system with the use of a zero phase error tracking controller and a feed forward compensator. Kobayashi et al. proposed a phase-stabilized servo controller in [34] where the structural modes of the piezoelectric actuator are used to compensate for errors from windage disturbance caused by the suspension vibration. Like the setup presented in this thesis, the fine piezoelectric actuator typically has stroke limits an order of magnitude lower than the coarse voice coil motor actuator. In [35], Herrmann et al. addressed the problem of fine actuator saturation by implementing an anti-windup scheme. Other authors have implemented state space based controllers for dual-stage feed drives. In [36] and [37], basic implementations of H_∞ and sliding mode controllers, respectively, were applied to the dual-stage configuration. She et al. [38], applied the equivalent input disturbance approach to a dual stage feed drive configuration. In [39] the coarse actuator was driven with an Adaptive Proximate Time Servomechanism and the fine actuator was controlled by a Composite Nonlinear Feedback controller to reduce the settling time. In this work the stiffness of the controller of the fine actuator was varied depending on the read head's proximity to the target. When the read head approaches the target the stiffness of the controller of the fine actuator increases, therefore increasing tracking precision.

Based on this literature survey it can be seen that many control laws exist for single stage feed drives but few techniques exist for configurations with multi-axis dual stage configurations. The dual stage configurations and subsequently control laws that are found in manufacturing literature are relatively limited, with no application of more sophisticated control algorithms on both the coarse and fine stages of actuation. Furthermore, the unique configurations of the works in multi-axis dual stage configurations do not necessarily apply to the configuration presented in this work.

2.4 Geometric Error Modeling, Identification, and Compensation

The modeling and identification of geometric errors began in the literature for the control of coordinate measuring machines (CMM). Since there are no process forces, software compensation of geometric errors in CMMs was a viable option. In one of the earliest examples of software compensation, Zhang et al. [40] modified feedback signals with a look up table in order to compensate for geometric errors. Similar techniques have been extrapolated to 3-axis CNC machine tools [41] where homogenous transformation matrices are used to map the axes geometric errors to the tooltip position errors. In [42] a 3D probe-ball and spherical test are developed to measure the link errors in the rotary table of a 5-axis CNC machine tool. The tool deviations caused by the axes geometric errors are then compensated in real-time by multiplying the inverse Jacobian matrix of the forward kinematic model with the tool deviation to obtain a compensating trajectory. Huang et al. also takes a similar approach in [43]. Alternatively, in [44], a method is proposed in which the trajectories for the A and C axis are modified to compensate for orientation geometric errors, then the X, Y, and Z axis are modified to compensate for position geometric error, therefore avoiding the recursive nature of simultaneously solving for compensation commands on all axes. However this assumes that moving the translational axes will not incur any orientation errors, which may not be necessarily true. In [45] the geometric errors are measured with ball-bar tests. The tooltip errors are mapped into the errors in the axes positions then added back to the axes positions to compensate for the errors. In [46], the geometric errors are first measured with a ball-bar. The positional deviations are corrected using the geometric error tables built into the CNC machine. However, positional deviations caused by rotational geometric errors are corrected by modifying the NC code since these errors are not available for compensation in a standard geometric error table in CNC machines. Xiang et al. [47] modeled the geometric errors of a 5-axis CNC machine tool using screw theory and then compensated them using inverse kinematics. At present, there is ongoing research on the geometric error compensation in 5-axis machine tools.

As it can be seen, in the above mentioned strategies, the geometric errors are compensated by modifying the toolpath with respect to the modeled geometric errors. However, comparatively less work has been done on the compensation of geometric errors with a secondary actuator. This may be a desirable configuration as there may be scenarios in which the trajectory of the major actuator cannot be modified. In literature there are a few examples that attempt to tackle this problem. A secondary magnetic bearing module has been used to compensate for the straightness errors of a single axis [48]. In another work, a two dimensional PZT actuator is used to compensate for the straightness error and positioning errors by sending the PZT actuator the inverted geometric error measured by a laser interferometer [49]. As it can be seen, the existing geometric error compensation techniques with the use of an external device are limited to a single axis and limited in general. Furthermore the standard geometric error compensation techniques mainly deal with configurations with analytic inverse kinematics, which allow for an analytic inverse differential solution. Since the rotary table has 6 degrees of freedom to compensate for the 3 translational errors, analytic inverse kinematic equations do not exist, and as a result, new techniques must be developed.

Chapter 3

Trajectory Generation for a 9-axis Micromachining Center

3.1 Overview

In robotics literature, the trajectory generation techniques are capable of resolving redundant degrees of freedom, but at present no works are able to do this in an optimal way, where the travel time is minimized while simultaneously respecting the velocity, acceleration, and jerk limits of the drives. Conversely, machine tool literature consists of trajectory generation techniques where the travel time is optimal while respecting the physical constraints of all drives involved, but have not been developed to resolve redundant degrees of freedom. In this chapter, a trajectory generation technique is presented where the strength of trajectory generation in robotics and machine tool literature are combined.

Given a computer aided design (CAD) geometry of the part to be manufactured, the computer aided manufacturing (CAM) system will generate a corresponding toolpath for the tool to follow, which will typically consist of desired tooltip positions and tool orientations at varying displacement intervals. In the case of the configuration presented, the trajectory generation algorithm must first decompose the desired tooltip positions and orientation into 9-axis commands which correspond to the desired tooltip position and orientation. Next it must schedule these commands in such a way that the velocity, acceleration, and jerk limits of all the axes are not violated while traversing the toolpath as fast as possible.

In order to do this, a forward kinematic model is first developed. Based on this kinematic model and the desired tooltip position and tool orientations, the redundancies are resolved numerically and corresponding axes positions are found at fixed displacement intervals along the toolpath. These axes positions are then scheduled in time with a feedrate optimizer, to be as fast as possible without violating the limits of the axes. Finally the feed profile is resampled, resulting in real-time position commands to be sent to the drives of the machine tool.

3.2 Forward Kinematics of the 9-axis Micromachining Center

Given the position commands of each of the drives it is possible to find the tooltip position and tool orientation using the forward kinematic model of the machine tool. In order to generate the forward kinematic model of a machine tool, first, two kinematic chains are formed with homogenous transformation matrices (HTM) based on the sequence of the drives [50]. In the configuration presented, the workpiece kinematic chain starts from an arbitrary base coordinate frame and ends at the workpiece, ${}^{B}\mathbf{T}_{w}$, and the tool kinematic chain starts from the same arbitrary base frame and ends at the tool holder, ${}^{B}\mathbf{T}_{t}$, as shown in Fig. 3.1.

With the kinematic chains defined, the inverse of the workpiece kinematic chain is multiplied with the tool kinematic chain to obtain the final transformation matrix, ${}^{w}\mathbf{T}_{t}$, which transforms positions defined in the tool coordinate into the workpiece coordinate frame. The first kinematic chain consists of the x-y table of the micromill and all the axes of the rotary table. As a result, based on this sequence, the transformation matrix to transform from the arbitrary base

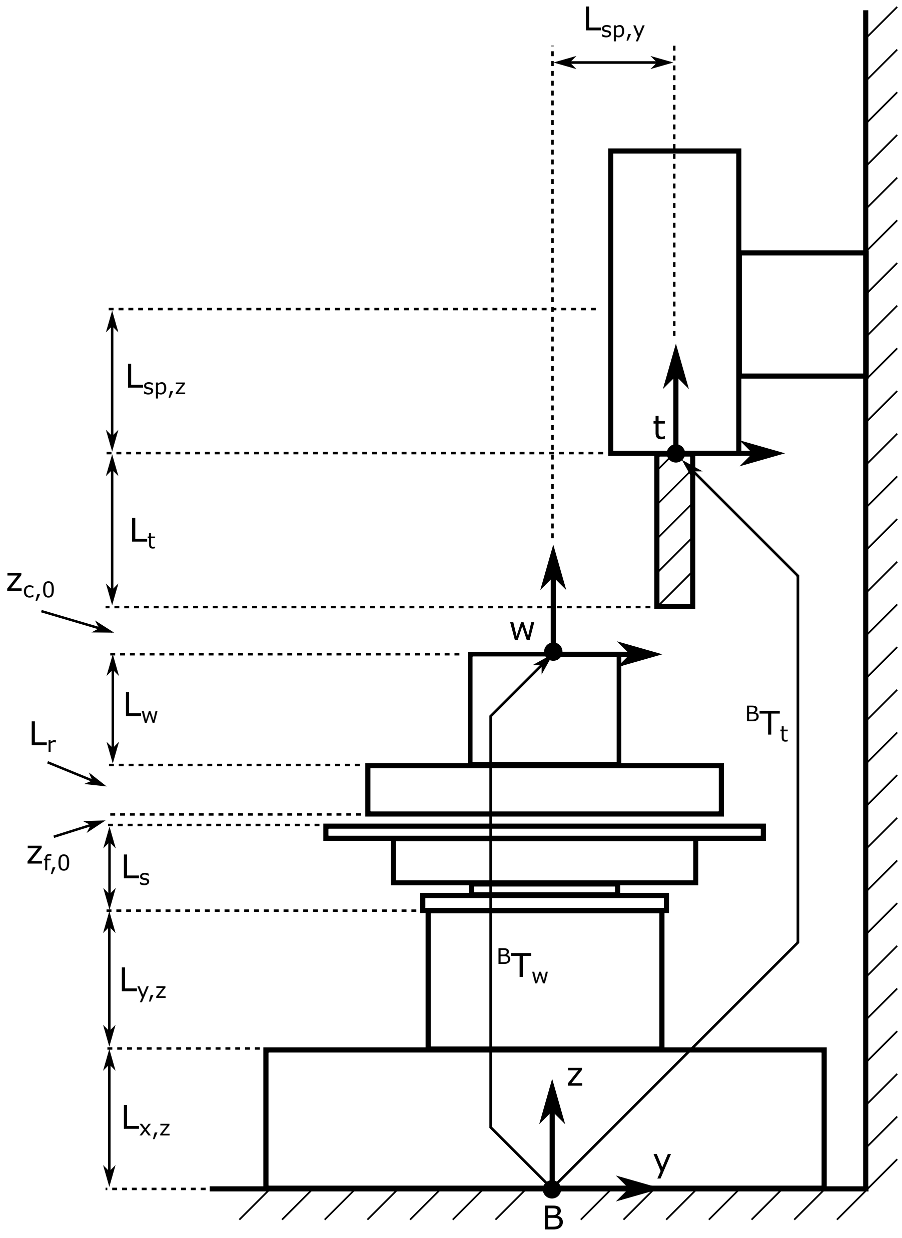


Figure 3.1: 9-axis micromachining center and corresponding axes of the machine tool

coordinate system to the workpiece coordinate frame is as follows:

$$\begin{aligned}
{}^{B}\mathbf{T}_{w,i} \;=\; & \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y_c \\ 0 & 0 & 1 & L_{y,z} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{B}\mathbf{T}_{y_c,i}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & x_c \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_{x,z} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{y_c}\mathbf{T}_{x_c,i}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_s \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{x_c}\mathbf{T}_{s,i}} \\
\times & \underbrace{\begin{bmatrix} 1 & 0 & 0 & x_f \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{s}\mathbf{T}_{x_f,i}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y_f \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{x_f}\mathbf{T}_{y_f,i}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z_f + z_{f0} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{y_f}\mathbf{T}_{z_f,i}} \\
\times & \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathsf{c}_{a_f} & -\mathsf{s}_{a_f} & 0 \\ 0 & \mathsf{s}_{a_f} & \mathsf{c}_{a_f} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{z_f}\mathbf{T}_{a_f,i}} \underbrace{\begin{bmatrix} \mathsf{c}_{b_f} & 0 & \mathsf{s}_{b_f} & 0 \\ 0 & 1 & 0 & 0 \\ -\mathsf{s}_{b_f} & 0 & \mathsf{c}_{b_f} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{a_f}\mathbf{T}_{b_f,i}} \\
\times & \underbrace{\begin{bmatrix} \mathsf{c}_{c_f} & -\mathsf{s}_{c_f} & 0 & 0 \\ \mathsf{s}_{c_f} & \mathsf{c}_{c_f} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{b_f}\mathbf{T}_{c_f,i}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_r \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{c_f}\mathbf{T}_{r,i}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_w \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{r}\mathbf{T}_{w,i}}
\end{aligned} \tag{3.1}$$

where $\mathsf{c}_q = \cos(q)$ and $\mathsf{s}_q = \sin(q)$, q is a general axis position, and ${}^{a}\mathbf{T}_{b,i}$ denotes a transformation matrix from coordinate frame a to coordinate frame b and the subscript i indicates an ideal homogenous transformation. It should be noted that in Eq.(3.1), the coordinate frames s, r, and w, denote the stator, rotor, and workpiece coordinate frames, respectively. Furthermore, x_c and y_c are the position commands of the micromill, x_f, y_f, z_f, a_f, b_f, and c_f are the position commands of the rotary table, $L_{x,z}$ is the height of the x_c-axis, $L_{y,z}$ is the height of the y_c-axis, L_s is the height of the stator, L_r is the height of the rotor, L_w is the height of the workpiece, and $z_{f,0}$ is the initial floating distance of the rotary table. All the offsets are static and shown in Fig. 3.1.

The second kinematic chain consists of the z-axis the micromill and the spindle offsets. As a result, the tool kinematic chain is defined as:

$$ {}^{B}\mathbf{T}_{t,i} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z_c + L_{0,z} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{B}\mathbf{T}_{z_c,i}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_{sp,y} \\ 0 & 0 & 1 & -L_{sp,z} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{z_c}\mathbf{T}_{t,i}} \tag{3.2} $$

where z_c is the z-axis position command of the micromill, $L_{sp,y}$ and $L_{sp,z}$ are the linear offset of the spindle, and $L_{0,z}$ is the initial offset of the z-axis of the micromill from the arbitrary base coordinate frame. The initial offset, $L_{0,z}$, can be defined as:

$$ L_{0,z} = L_{x,z} + L_{y,z} + L_s + z_{f,0} + L_r + L_w + z_{c,0} + L_{sp,z} + L_t \tag{3.3} $$

where $z_{c,0}$ is the initial position of the z-axis of the micromill and L_t is the length of the cutting tool. With both transformation matrices originating at the arbitrary base coordinate frame, the transformation matrix from the tool coordinate frame to the workpiece coordinate frame is found by multiplying the inverse of Eq. (3.1) with Eq. (3.2) as follows:

$$ {}^{w}\mathbf{T}_{t,i} = \left({}^{B}\mathbf{T}_{w,i}\right)^{-1} {}^{B}\mathbf{T}_{t,i} \tag{3.4} $$

If the position of the tooltip is defined in the tool coordinate frame, then it is possible to find its position relative to the workpiece with the transformation defined in Eq. (3.4). Relative to the tool coordinate frame, the tooltip position is the exposed length of the cutting tool and the orientation is always lined up with the z-axis of the tool coordinate frame. As a result, the tool can be defined in the tool coordinate frame as ${}^{t}\mathbf{t}_p = [0\;0\;L_t]^T$ and ${}^{t}\mathbf{t}_o = [0\;0\;1]^T$ for the tooltip position and tool orientation, respectively. Given the transformation matrix in Eq. (3.4), the tooltip position and tool orientation can be transformed into the workpiece coordinate frame as follows:

$$\begin{bmatrix} P_x & O_i \\ P_y & O_j \\ P_z & O_k \\ 1 & 0 \end{bmatrix} = \mathbf{T} \begin{bmatrix} {}^{t}\mathbf{t}_p & {}^{t}\mathbf{t}_o \\ 1 & 0 \end{bmatrix} \tag{3.5}$$

Expanding the matrix equation defined in Eq. (3.5), the forward kinematic model gives the tooltip position and tool orientation defined as a function of the positions of the 9 drives as follows:

$$\begin{aligned} P_x(\mathbf{q}) \;=\; & (-L_{sp,y} - y_c - y_f)(\mathrm{c}_{a_f}\mathrm{s}_{c_f} + \mathrm{c}_{c_f}\mathrm{s}_{a_f}\mathrm{s}_{b_f}) \\ & +(L_r + L_w + z_{c,0} + z_c - z_f)(\mathrm{s}_{a_f}\mathrm{s}_{c_f} - \mathrm{c}_{a_f}\mathrm{c}_{c_f}\mathrm{s}_{b_f}) \\ & +\mathrm{c}_{b_f}\mathrm{c}_{c_f}(-x_c - x_f) \end{aligned} \tag{3.6}$$

$$\begin{aligned} P_y(\mathbf{q}) \;=\; & (-L_{sp,y} - y_c - y_f)(\mathrm{c}_{a_f}\mathrm{c}_{c_f} - \mathrm{s}_{a_f}\mathrm{s}_{b_f}\mathrm{s}_{c_f}) \\ & +(L_r + L_w + z_{c,0} + z_c - z_f)(\mathrm{c}_{c_f}\mathrm{s}_{a_f} + \mathrm{c}_{a_f}\mathrm{s}_{b_f}\mathrm{s}_{c_f}) \\ & +(x_c + x_f))\mathrm{c}_{b_f}\mathrm{s}_{c_f} \end{aligned} \tag{3.7}$$

$$\begin{aligned} P_z(\mathbf{q}) \;=\; & (-x_c - x_f)\mathrm{s}_{b_f} - L_{wz} - L_r \\ & +(L_r + L_w + z_{c,0} + z_c - z_f)\mathrm{c}_{a_f}\mathrm{c}_{b_f} \\ & +(L_{sp,y} + y_c + y_f))\mathrm{c}_{b_f}\mathrm{s}_{a_f} \end{aligned} \tag{3.8}$$

$$O_i(\mathbf{q}) = \mathrm{s}_{a_f}\mathrm{s}_{c_f} - \mathrm{c}_{a_f}\mathrm{c}_{c_f}\mathrm{s}_{b_f} \tag{3.9}$$

$$O_j(\mathbf{q}) = \mathrm{c}_{c_f}\mathrm{s}_{a_f} + \mathrm{c}_{a_f}\mathrm{s}_{b_f}\mathrm{s}_{c_f} \tag{3.10}$$

$$O_k(\mathbf{q}) = \mathrm{c}_{a_f}\mathrm{c}_{b_f} \tag{3.11}$$

where $\mathbf{q} = [x_c, y_c, z_c, x_f, y_f, z_f, a_f, b_f, c_f]^T$, P_x, P_y, and P_z are the x, y, and z position of the tooltip in the workpiece coordinate frame, and O_i, O_j, and O_k are the orientation of the tool about the x, y, and z axis of the workpiece coordinate frame. In the next section, the Jacobian of the forward kinematics, as defined in Eq. (3.6) to (3.11), along with the desired toolpath is used to resolve the redundant degrees of freedom.

3.3 Redundancy Resolution of the 9-axis Micromachining Center

Typically, machining toolpaths from the CAM system are specified as the tooltip position and tool orientation with respect to the workpiece, also known as cutter location (CL) data. The CL data from the CAM system is first fit to a jerk continuous B-spline as a function of tangential displacement , s, to preserve third order continuity [51] in order to avoid exciting the structural modes of the machine tool:

$$\mathbf{R}(s) = [P_x(s), P_y(s), P_z(s), O_i(s), O_j(s), O_k(s)]^T \tag{3.12}$$

In traditional 3-axis and 5-axis machine tools, analytic inverse kinematic equations can be used to find the corresponding axes positions given $\mathbf{R}(s)$. However, since the 9-axis micromachining center has more axes than necessary, the system is under constrained and analytic inverse kinematic solutions do not exist. In this work, numerical techniques from robotics literature [6] [7] are adapted to resolve the under constrained system. In order to generate trajectories that resolve the redundancy of the micromachining center, axes configurations, which correspond to the desired tooltip position and tool orientation, at fixed displacement intervals along the toolpath are first found. With the forward kinematic equations, it is possible to find the Jacobian matrix, which is the differential of the tooltip position and tool orientation with respect to axes positions:

$$J = \frac{d\mathbf{R}}{d\mathbf{q}} = \begin{bmatrix} \frac{dP_x}{dx_c} & \cdots & \frac{dP_x}{dc_f} \\ \vdots & \ddots & \vdots \\ \frac{dO_k}{dx_c} & \cdots & \frac{dO_k}{dc_f} \end{bmatrix}_{6\times 9} \tag{3.13}$$

Given differential axes positions, it is possible to find the differential tooltip positions and tool orientations, as follows:

$$d\mathbf{R} = J d\mathbf{q} \tag{3.14}$$

However, since the objective is to find $\mathbf{q}$ from $\mathbf{R}(s)$, the inverse of J is required. Since J is not a square matrix it is not possible to simply multiple $d\mathbf{R}$ with the inverse of J. Alternatively, as originally proposed in [6], the Moore-Penrose inverse, defined as:

$$J^{\dagger} = J^{T}\left(JJ^{T}\right)^{-1} \tag{3.15}$$

can be used to find the differential axes positions. By multiplying the Moore-Penrose inverse with a differential tooltip position and tool orientation, it is possible to find the corresponding differential axes position. Since the spline describing the tooltip position and tool orientation, $\mathbf{R}(s)$, is a function of tangential displacement, s, it can be differentiated with respect to s and multiplied with $J^{\dagger}$ to get the axes positions differentiated with respect to displacement as follows:

$$\frac{d\mathbf{q}}{ds} = J^{\dagger}\frac{d\mathbf{R}}{ds} \tag{3.16}$$

It should be noted that an infinite range of differential solutions exist and Eq. (3.16) represents the least squares solution of the under constrained problem. This corresponds to the solution with the lowest average axes differential with respect to displacement at each step or more formally:

$$\begin{aligned} &\text{minimize} \quad \left|\left|\frac{d\mathbf{q}}{ds}\right|\right| \\ &\text{subject to} \quad d\mathbf{R} = J d\mathbf{q} \end{aligned} \tag{3.17}$$

However, since our trajectory has requirements beyond the least norm solution, Eq. (3.16) is augmented to fulfill other requirements such as avoiding stroke limits and singularities. Furthermore, the differential solution must be numerically integrated at each step in order to obtain position commands at fixed displacement intervals.

3.4 Avoiding Stroke Limits of the Drives

As mentioned, Eq. (3.16) represents a single solution in an infinite range of solutions to Eq. (3.14). As a result Eq. (3.16) can be augmented to fulfill other criteria by using the nullspace of the Jacobian matrix J in the following way:

$$\frac{d\mathbf{q}}{ds} = J^{\dagger}\frac{d\mathbf{R}}{ds} + \beta\left(I - JJ^{\dagger}\right)\nabla H \tag{3.18}$$

where $\left(I - JJ^{\dagger}\right)$ is the nullspace of the Jacobian, β is a constant gain, and H is a cost function to be minimized. Any projection onto the nullspace does not effect the position of the end effector, so joint configurations can be selected that minimize the cost function defined in H without affecting the end effector's position. Due to the highly varying stroke limits of each of the two systems, it is important to ensure that the stroke limits of each of the drives are respected when the trajectories are generated. In order to constraint the generated axes commands, the cost function is defined in a way that position commands close to the stroke limits of the drives are penalized:

$$H(\mathbf{q}) = \frac{x_c^2}{\left(x_{c,max}^2 - x_c^2\right)^2} + \cdots + \frac{c_f^2}{\left(c_{f,max}^2 - c_f^2\right)^2} \tag{3.19}$$

where q_{max} is the stroke limit of the respective axis. As it can be seen, the cost function approaches infinity as the axes positions approach the stroke limits. Equation (3.18) projects the gradient of H, which is defined as:

$$\nabla H = \left[\frac{2x_{c,max}^2 x_c}{\left(x_{c,max}^2 - x_c^2\right)^2}, \ldots, \frac{2c_{f,max}^2 c_f}{\left(c_{f,max}^2 - c_f^2\right)^2}\right] \tag{3.20}$$

on to the nullspace Jacobian, thereby selecting an axes configuration that gives the desired tooltip position and tool orientation, while respecting the stroke limits of the drives. By substituting Eq. (3.20) into Eq. (3.18), the solution works similar to a gradient descent optimization, where the solution selects the configuration that minimizes the cost function defined by H. It should be noted the solution divides its efforts between attempting to minimize the norm of the axes differentials for all the axes and minimizing the cost function H.

To show this, a trajectory for a circle of radius 4 [mm] on the x-y plane is generated using only the x-y axis of the micromill and rotary table. Figures 3.2 and 3.3 shows trajectories generated by the algorithm without and with the stroke constraints, respectively. Figure 3.4 compares the norm of the reference commands for the constrained and unconstrained configurations. As it can be seen, without the constraint, as shown in Fig. 3.2, the fine and coarse actuator have the same motion profile, resulting in the lowest average axes movements as shown in the norm of the motion commands in Fig. 3.4. However this would not be possible as the fine actuators only have a stroke limit of 100 [μm]. When the constraints are included, the fine actuator is constrained to its stroke limits while the coarse actuator takes up the remainder of the trajectory, as shown in Fig. 3.3, resulting in higher average axes movements as shown in Fig. 3.4.

The strength of each portion of the solution is dictated by the gain β. A larger β results in a solution that prefers to keep the position commands away from the stroke limits but may not necessarily have the lowest average axes movement. Conversely, a small β reduces the average axes movement but will have configurations that are closer to the stroke limits. It should be noted that if β is too small, Eq. (3.18) may result in solutions that violate the stroke limits.

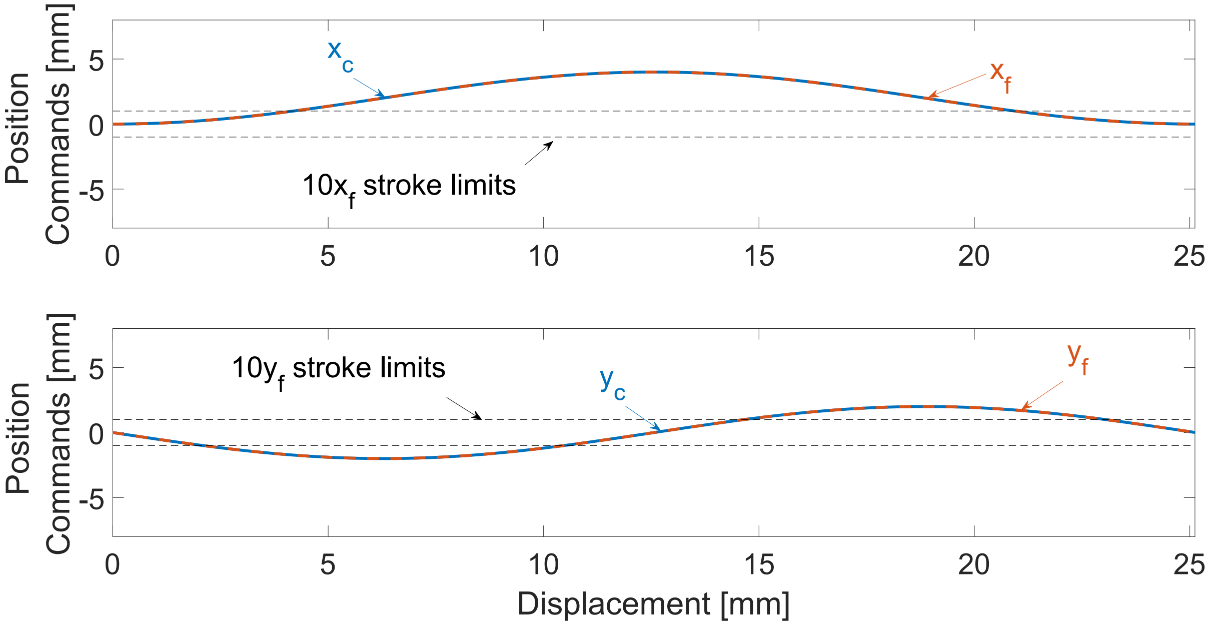


Figure 3.2: Generated trajectories without constraint cost function

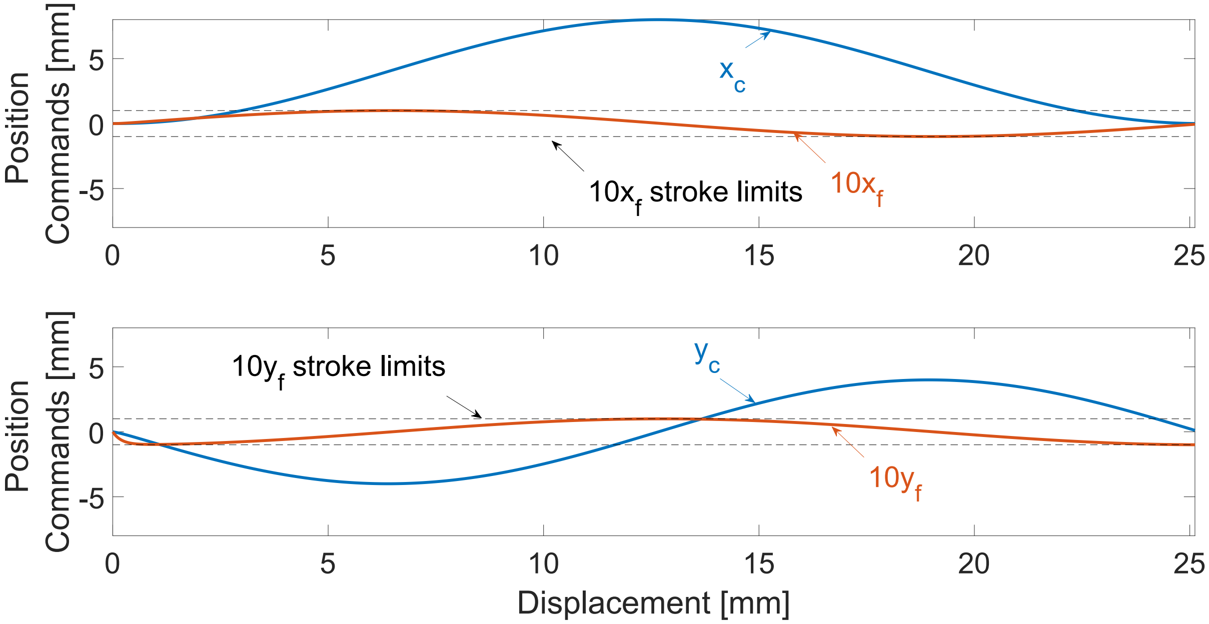


Figure 3.3: Generated trajectories with constraint cost function

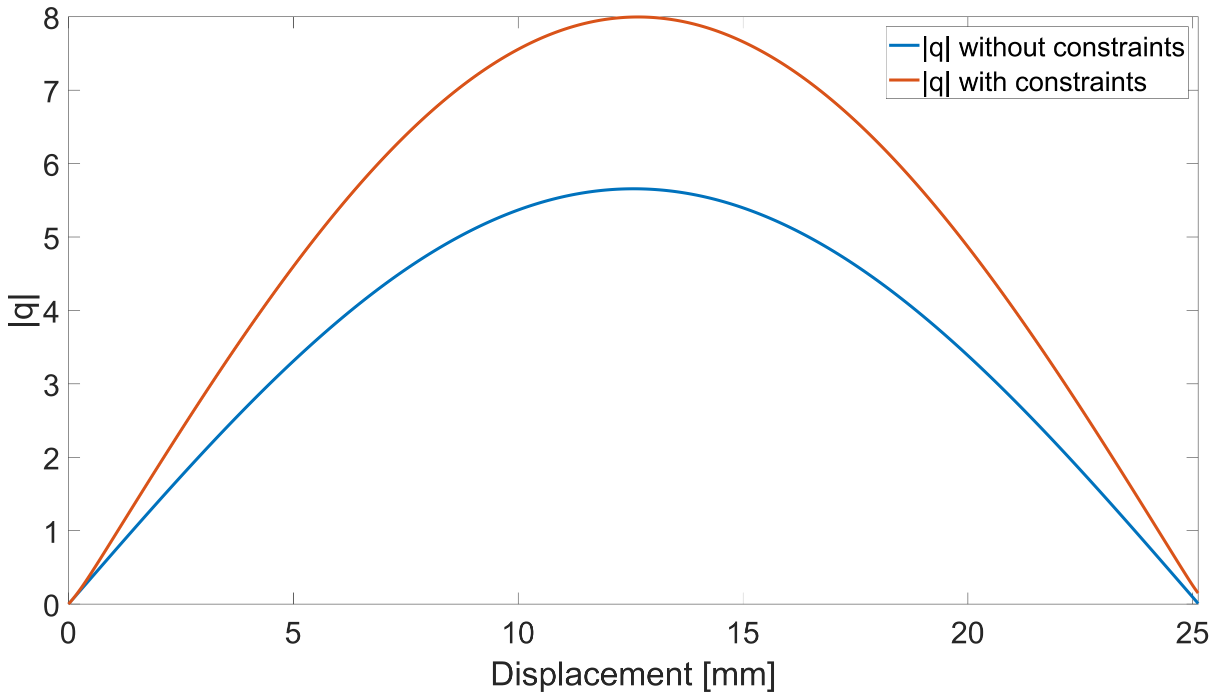


Figure 3.4: The norms of $\frac{d\mathbf{q}}{ds}$ with and without constraint cost function

3.5 Singularity Avoidance with augmentation of Jacobian Matrix

Due to the non-Cartesian movements of the 9-axis micromachining center, it is possible to end up in a singular configuration where the machine tool losses actuation capabilities in a direction. Mathematically, it results in the Jacobian matrix becoming degenerate, and as a result, the Moore-Penrose inverse cannot be found and Eq. (3.18) will not have a solution. In order to avoid this scenario, the Jacobian matrix can be augmented by adding a diagonal matrix of small constants [10] so that a solution can always be found as follows:

$$J^* = J^T \left(JJ^T + \mu I\right)^{-1} \tag{3.21}$$

The trade off for guaranteeing a solution is the introduction of an error into the solution, since the Moore-Penrose inverse has been modified. However, an augmented solution is only necessary in the vicinity of a singularity. As a result, the Moore-Penrose inverse is conditionally modified by varying the magnitude of the small constant μ in the following way:

$$\mu = \begin{cases} \mu_0 \left(1 - \frac{w}{w_0}\right)^2, & \text{if } w < w_0 \\ 0, & \text{otherwise} \end{cases} \tag{3.22}$$

where $w = \sqrt{\det(JJ^T)}$ is manipulability, which corresponds to how close the machine is to a singular configuration. w_0 is the threshold for manipulability in which the modification constant μ starts taking on a non-zero value. Furthermore, μ_0 corresponds to the maximum amount of deviation the solution is allowed to have from the unmodified solution. If w_0 and μ_0, are set too low the solution may still be singular, resulting in a degenerate solution. In contrast, if w_0 and μ_0 are too large, then the deviation from the desired path may be too large, violating user defined tolerances. The different effects of different sizes of w_0 and μ_0 can be seen in Fig. 3.5 and Fig. 3.6, where $w_0 = \mu_0 = 1$ in Fig. 3.5 and $w_0 = \mu_0 = 10^{-6}$ in Fig. 3.6. As a result w_0 and μ_0 are tuning parameters which depend on the accuracy constraints.

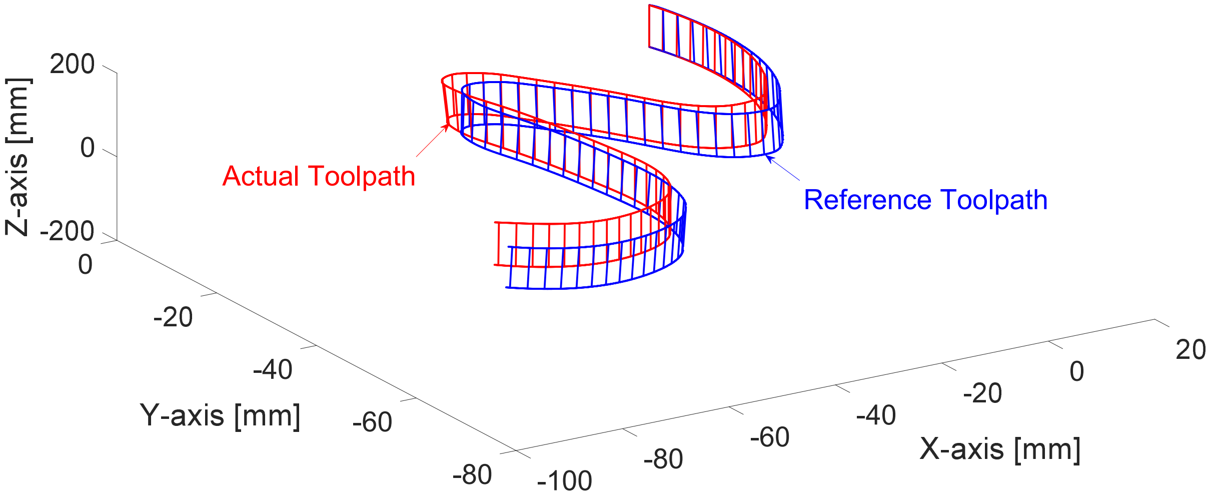


Figure 3.5: Desired toolpath and generated tool path with $w_0 = \mu_0 = 1$

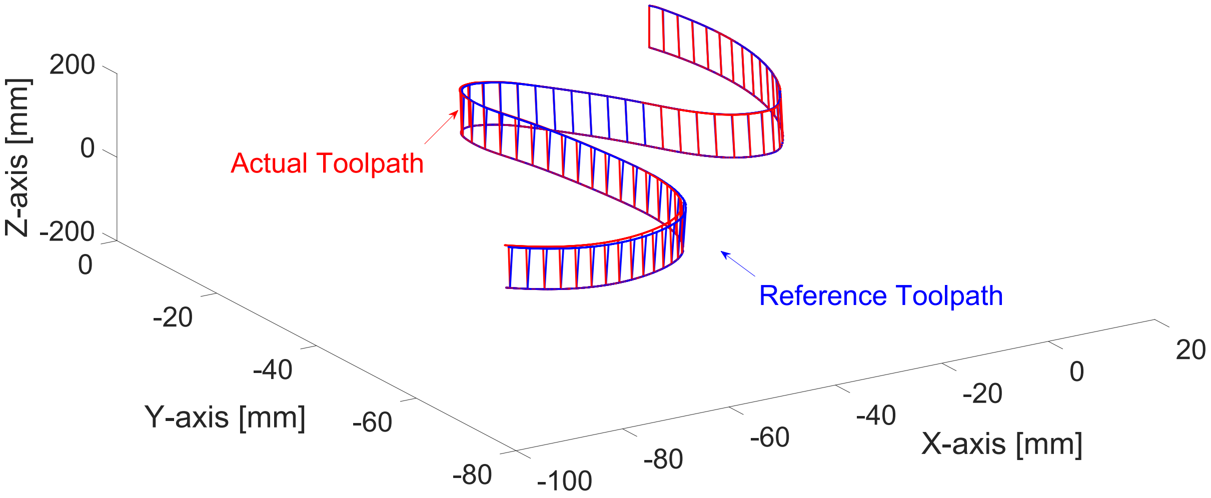


Figure 3.6: Desired toolpath and generated tool path with $w_0 = \mu_0 = 10^{-6}$

3.6 Numerical Integration of Differential Solution to get Position Commands

Though the solution is capable of avoiding the stroke limit of drives and singularities, the solution only outputs a differential, $\frac{d\mathbf{q}}{ds}$. In order to obtain axes positions, the solution must be integrated over displacement, s, as follows:

$$\mathbf{q}(s) = \int_0^s \frac{d\mathbf{q}}{ds} ds \tag{3.23}$$

However, the analytic integration $\frac{d\mathbf{q}}{ds}$ is not straightforward and the integration is performed numerically instead. The length of the whole toolpath, S, is first found with Simpson's method [52], then divided into N intervals giving us the fixed displacement intervals $\Delta s = S/N$. Through numerical integration, the axes positions $\mathbf{q}[n]$ are found at the displacements $s[n] = \Delta s \times n$ where $n = 1, 2, 3, \ldots, N-1, N$. Several works [12] have analyzed various ways to perform the numerical integration and as expected, the 4th Order Runge Kutta method provided the best results. Since the initial joint configuration is known, and it is known that there is no displacement at the beginning of the tool path, the initial values can be set as $\mathbf{q}[0] = \mathbf{q}_0$ and $s[0] = 0$. With these initial values, the numerical integration can be performed in the following way:

$$\begin{aligned}
\mathbf{q}[n+1] &= \mathbf{q}[n] + (k_1 + 2k_2 + 2k_3 + k_4)\tfrac{\Delta s}{6} \\
k_1 &= \frac{d\mathbf{q}\left(s[n], \mathbf{q}[n]\right)}{ds} \\
k_2 &= \frac{d\mathbf{q}\left(s[n] + \frac{\Delta s}{2}, \mathbf{q}[n] + \frac{\Delta s}{2} k_1\right)}{ds} \\
k_3 &= \frac{d\mathbf{q}\left(s[n] + \frac{\Delta s}{2}, \mathbf{q}[n] + \frac{\Delta s}{2} k_2\right)}{ds} \\
k_4 &= \frac{d\mathbf{q}\left(s[n] + \Delta s, \mathbf{q}[n] + \Delta s k_3\right)}{ds}
\end{aligned} \tag{3.24}$$

which gives N axes positions, $\mathbf{q}[n]$, that correspond to the N desired tool position and orientation $\mathbf{R}(s[n])$ at displacements $s[n]$. Unlike the majority of works in robotics literature that employ a similar technique, the commands are generated with respect to displacement, allowing for further optimization over time. The 4th Order Runge Kutta integration is repeated until the end of the toolpath is reached, resulting in known axes positions at fixed displacement intervals.

3.7 Closed loop correction of deviations from numerical integration

Due to the numerical nature of the integration algorithm, a small amount of numerical drift occurs at each step, causing the corresponding tooltip position and tool orientation from the axes positions $\mathbf{q}[n]$ to deviate from the desired tooltip position and tool orientation specified by $\mathbf{R}(s)$. As a result, a corrective action is introduced into the differential solution in Eq. (3.18) [13]. Given the solved axes position $\mathbf{q}[n]$ at a given displacement $s[n]$, it is possible to find the resultant tooltip position and tool orientation with the forward kinematics as described in Eq. (3.6) to (3.11). For brevity, the forward kinematic equations are grouped into a single vector function:

$$\mathbf{f}(\mathbf{q}[n]) = \begin{bmatrix} P_x\left(\mathbf{q}[n]\right) \\ P_y\left(\mathbf{q}[n]\right) \\ P_z\left(\mathbf{q}[n]\right) \\ O_i\left(\mathbf{q}[n]\right) \\ O_j\left(\mathbf{q}[n]\right) \\ O_k\left(\mathbf{q}[n]\right) \end{bmatrix} \tag{3.25}$$

As result, the error between the desired tooltip position and orientation and the result of the differential solution can be found as follows:

$$\mathbf{e}[n] = \mathbf{R}(s[n]) - \mathbf{f}(\mathbf{q}[n]) \tag{3.26}$$

Given the error $\mathbf{e}[n]$, it is possible to include closed loop corrective action into Eq. (3.18) so axes configurations can be selected to correct the numerical deviation caused by the numerical integration algorithm as follows:

$$\frac{d\mathbf{q}}{ds} = J^{*}\left(K_e\mathbf{e}[n] + \frac{d\mathbf{R}}{ds}\right) + \beta(\mathbf{I} - J^{*}J)\nabla\mathbf{H} \tag{3.27}$$

where K_e is a scalar gain. As a result, similar to a controller, the differential solution seeks to minimize the error at each subsequent step. It should be noted that, the size of the scalar gain is limited by the stability of the solution. Should too high of a gain be set, the solution will become unstable. To demonstrate its effectiveness, Fig. 3.8 and Fig. 3.7 show the errors of the Runge Kutta numerical integration algorithm, with and without the closed loop correction implemented, respectively.

From Fig.3.8 it should be noted that the 4th Order Runge Kutta integration method is able to keep the numerical errors in the range of 10^{-11} [mm] and 10^{-10} [rads] which is well below the tolerance in most machining applications. However, it is important to note that since there is no corrective action for numerical drift, as the toolpath gets longer, numerical errors may be incurred at each step causing violation at some point on the toolpath. From Fig. 3.8, the closed loop corrective action corrects for the numerical drift resulting in a numerical error of 10^{-13} [mm] and 10^{-12} [rads], at least an order of magnitude lower than without the corrective action. Furthermore, error is not incurred at each step, ensuring that the algorithm will not violate the tolerance for longer toolpaths.

3.8 Full Redundancy Resolution Algorithm

The full redundancy resolution algorithm incorporates numerous elements, including cost functions, singularity avoidance, numerical integration, and closed loop corrective action. For brevity:

$$\frac{d\mathbf{q}}{ds}\left(s[n], \mathbf{q}[n]\right) = \mathbf{g}\left(s[n], \mathbf{q}[n]\right) \tag{3.28}$$

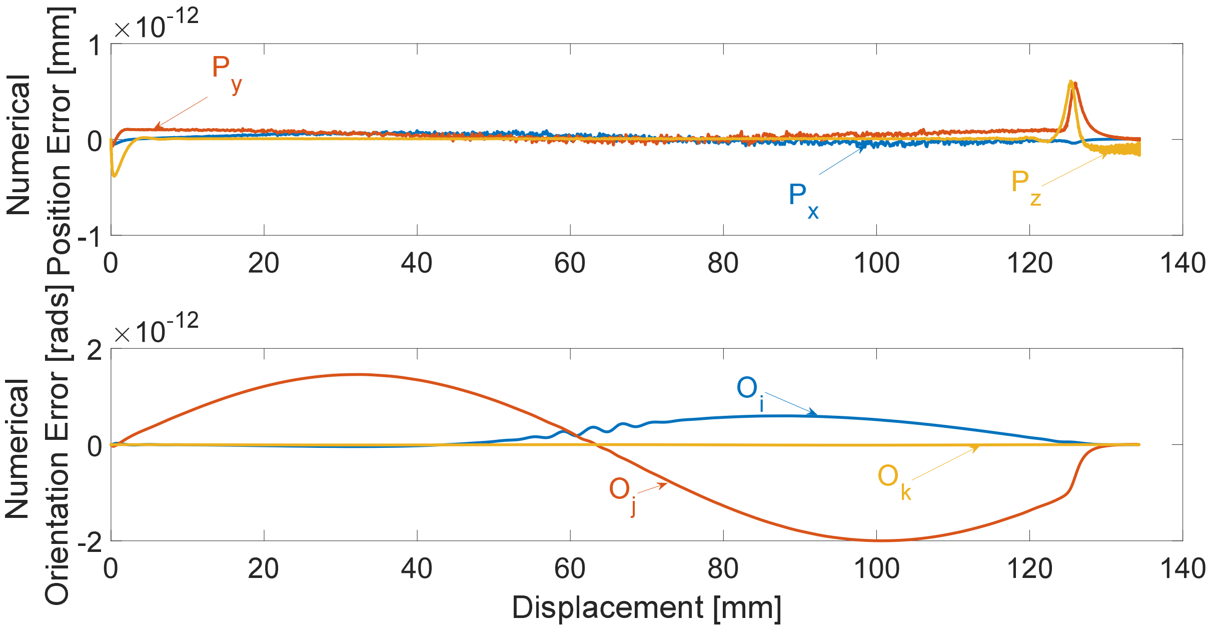


Figure 3.7: Numerical errors of the redundancy resolution algorithm with closed loop correction

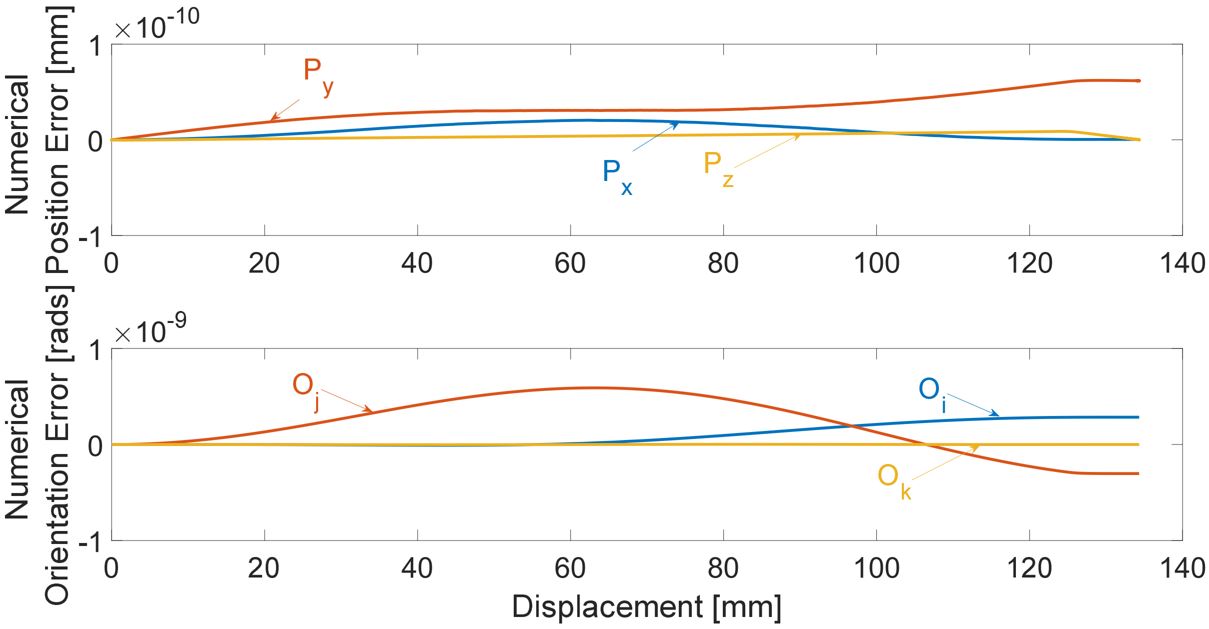


Figure 3.8: Numerical errors of the redundancy resolution algorithm without closed loop correction

Algorithm 1: Full kinematic algorithm

```
function Kinematics ($\mathbf{s}, \mathbf{Q}$);
Input  : $\mathbf{R}(s)$, $S$, $N$
Output: $\mathbf{s} = [s[0], s[1], \ldots, s[N-1], s[N]]$
        $\mathbf{Q} = [\mathbf{q}[0], \mathbf{q}[1], \ldots, \mathbf{q}[N-1], \mathbf{q}[N]]$
$s[0] = 0$;
$\mathbf{q}[0] = \mathbf{q}_0$;
$\Delta s = S/N$;
$n = 0$;
for $n \leftarrow 0$ to $N-1$ do
    $k_1 = \mathbf{g}(\mathbf{q}[n], s[n])$;
    $k_2 = \mathbf{g}(\mathbf{q}[n] + k_1 \Delta s/2, s[n] + \Delta s/2)$;
    $k_3 = \mathbf{g}(\mathbf{q}[n] + k_2 \Delta s/2, s[n] + \Delta s/2)$;
    $k_4 = \mathbf{g}(\mathbf{q}[n] + \Delta s k_3, s[n] + \Delta s)$;
    $\mathbf{q}[n+1] = \mathbf{q}[n] + (k_1 + 2k_2 + 2k_3 + k_4)\Delta s/6$;
    $s[n+1] = s[n] + \Delta s$;
    $n = n + 1$;
end
```

With this definition, all the elements of the redundancy resolution algorithm are incorporated as shown in Algorithm 1.

3.9 Feedrate Optimization with Redundancy Resolution

The redundancy resolution algorithm is able to generate differential solutions to the under constrained problem with respect to displacement. As a result, with the numerical integration algorithm, we have the axes positions ,$\mathbf{q}[n]$,and axes differentials, $\frac{d\mathbf{q}[n]}{ds}$, at displacements $s[n]$. However, the axes commands have not been scheduled with respect to time, and further optimization is required. When scheduling the feed profile, the objective is to traverse the toolpath as fast as possible without violating the physical limits of the drives, or more formally:

$$\begin{aligned} \text{minimize} \quad & \int_0^S \frac{1}{\dot{s}(s)} ds \\ \text{subject to} \quad & \dot{\mathbf{q}} < \dot{\mathbf{q}}_{\text{max}} \\ & \ddot{\mathbf{q}} < \ddot{\mathbf{q}}_{\text{max}} \\ & \dddot{\mathbf{q}} < \dddot{\mathbf{q}}_{\text{max}} \end{aligned} \tag{3.29}$$

where $\dot{s}(s)$ is the feedrate defined as a function of the displacement, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, and $\dddot{\mathbf{q}}$ are the axes velocity, acceleration, and jerk, respectively, and $\dot{\mathbf{q}}_{\text{max}}$, $\ddot{\mathbf{q}}_{\text{max}}$, and $\dddot{\mathbf{q}}_{\text{max}}$ are the axes velocity limits, acceleration limits, and jerk limits, respectively. By minimizing the reciprocal of the feed profile over the length of the toolpath, the result is a feed profile that travels the toolpath as fast as possible. The feedrate is defined as a jerk continuous B-spline of the form:

$$\dot{s}(s) = \sum_{i=0}^{\Theta} N_{i,p}(\vartheta) P_i \tag{3.30}$$

where $N_{i,p}(\vartheta)$ are the B-spline basis functions, P_i are the control points, $\vartheta = s/S$ is the spline parameter equivalent to the displacement normalized by the total toolpath length, p is the degree of the spline, and Θ is the total of number of control points [53]. The number of control points is selected by considering the speed and accuracy of the optimization algorithm. With more control points, there are more degrees of freedom allowing for a more optimal feedrate but since the optimization algorithm is numerical, the computational load is higher. The control points are initialized as follows:

$$\begin{aligned} P_1 &= 0 \\ [P_2, P_3, \ldots, P_{\Theta-2}, P_{\Theta-1}] &= f/2 \\ P_\Theta &= 0 \end{aligned} \tag{3.31}$$

where f is the initial user defined feedrate. This corresponds to stationary start and stop conditions, with an initialized feedrate of half of the desired feedrate. It should be noted that the selection of $f/2$ for the initial feedrate is arbitrary since the optimization algorithm will change the feedrate. The knot vector is uniformly distributed and has $r = \Theta + p + 1$ knots. Since the feedrate starts at P_1 and ends at P_Θ, the first 3 knots of the knot vector are 0 and the last 3 knots of the knot vector are 1. The remaining $r - 6$ knots are then evenly distributed between 0 and 1. The knot vector can be assigned as follows:

$$\vartheta = [0, 0, 0, \Delta\vartheta, 2\Delta\vartheta, \ldots, (r-6)\Delta\vartheta, 1, 1, 1] \tag{3.32}$$

where $\Delta\vartheta = 1/(r-5)$. Next, the constraints are defined as a function of the feed profile and the output of the kinematic module as described in Algorithm 1. The axes velocity $\dot{\mathbf{q}}$, acceleration $\ddot{\mathbf{q}}$, and jerk $\dddot{\mathbf{q}}$, equations are expressed as a function of the axes differentials and the feedrate at the N displacement intervals as follows:

$$\dot{\mathbf{q}}[n] = \frac{d\mathbf{q}[n]}{ds}\dot{s}(s[n]) \tag{3.33}$$

$$\ddot{\mathbf{q}}[n] = \frac{d^2\mathbf{q}[n]}{ds^2}\dot{s}^2(s[n]) + \frac{d\mathbf{q}[n]}{ds}\ddot{s}(s[n]) \tag{3.34}$$

$$\dddot{\mathbf{q}}[n] = \frac{d^3\mathbf{q}[n]}{ds^3}\dot{s}^3(s[n]) + 3\frac{d^2\mathbf{q}[n]}{ds^2}\dot{s}(s[n])\ddot{s}(s[n]) + \frac{d\mathbf{q}[n]}{ds}\dddot{s}(s[n]) \tag{3.35}$$

where the second and third geometric derivatives, $\frac{d^2\mathbf{q}[n]}{ds^2}$ and $\frac{d^3\mathbf{q}[n]}{ds^3}$, can be found be differentiating Eq. (3.27). A set of constraint equations, $\mathbf{C}(s[n])$, which correspond to the physical limits of the drives is found by normalizing $\dot{\mathbf{q}}[n]$, $\ddot{\mathbf{q}}[n]$, and $\dddot{\mathbf{q}}[n]$ with the axes limits:

$$\mathbf{C}(s[n]) = \left[\frac{|\dot{\mathbf{q}}[n]|}{\dot{\mathbf{q}}_{\max}}, \frac{|\ddot{\mathbf{q}}[n]|}{\ddot{\mathbf{q}}_{\max}}, \frac{|\dddot{\mathbf{q}}[n]|}{\dddot{\mathbf{q}}_{\max}}\right]^T \leq \mathbf{1} \tag{3.36}$$

As a result, in order to ensure the limits of the drives are not violated when traversing the toolpath, the feed profile must be selected to ensure that the constraint equations defined in Eq. (3.36) are not greater than 1. Since the constraints are functions of the feedrate and its time derivatives, the optimization problem is a non-linear problem and must be solved with a non-linear optimizer. In order to do this, the problem must be phrased as a minimization problem as follows:

$$\begin{aligned} \underset{P_i}{\text{minimize}} \quad & \int_0^S \frac{1}{\sum_{i=0}^{\Theta} N_{i,p}(u)P_i} ds \\ \text{subject to} \quad & \mathbf{C}(s[n]) \leq 1,\ n = 0, \ldots, N. \end{aligned} \tag{3.37}$$

The reciprocal of the feedrate of the spline is minimized by modifying the control points, P_i, while checking the constraints at $\mathbf{C}(s[n])$. Since the kinematic solution is numerical, $\frac{d\mathbf{q}}{ds}$, $\frac{d^2\mathbf{q}}{ds^2}$ and $\frac{d^3\mathbf{q}}{ds^3}$ are only known at $s[n]$ displacements. As a result, the constraints are only evaluated at the known N displacements as specified in Eq. (3.37). Since the problem defined by Eq. (3.37) is a constrained non-linear optimization problem it is solved with a non-linear convex optimization solver found in MATLAB's optimization tool box. In this work the Active Set algorithm is selected as the optimization algorithm. Further details of this optimization algorithm can be found in [5].

Once an optimized feed profile is found, displacements are scheduled with respect to time. Since the feedrate is a function of displacement, the displacements with respect to time are found with a Taylor Expansion:

$$s[k+1] = s[k] + \dot{s}(s[k])T_s + \frac{\ddot{s}(s[k])}{2}T_s^2 \tag{3.38}$$

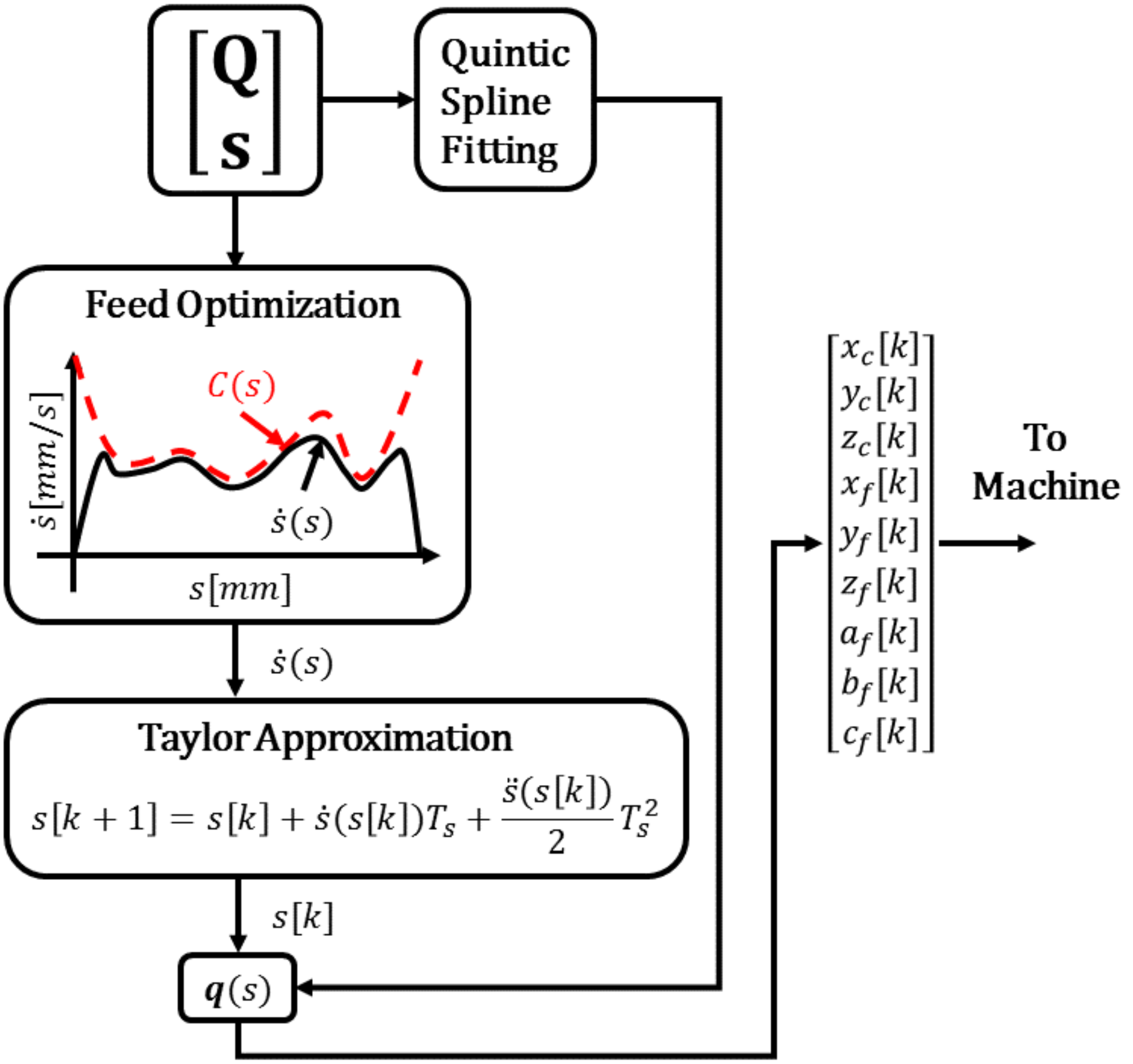
where $k = 1, 2, \ldots, K-1, K$ where K is the total number of time steps to traverse the toolpath. Equation (3.38) gives the desired tool path displacements $s[k]$ at time $k \times T_s$, based upon the feedrate from the optimization algorithm. However, the axes positions $\mathbf{q}$ are still only known at uniformly spaced intervals $\triangle s$ based on Eq. (3.24). To get a jerk continuous representation of the axes positions as a function of displacement, the joint positions are interpolated with septic polynomials:

$$\mathbf{q}_n(s) = \mathbf{A}_n s^7 + \mathbf{B}_n s^6 + \mathbf{C}_n s^5 + \mathbf{D}_n s^4 + \mathbf{E}_n s^3 + \mathbf{F}_n s^2 + \mathbf{G}_n s + \mathbf{H}_n \tag{3.39}$$

where the subscript n indicates a polynomial connecting the joint positions at $s[n]$ to the joint positions at $s[n+1]$. Septic polynomials are selected since there are 8 boundary conditions including the position, velocity, acceleration, and jerks at the start and stop points of the polynomial. Since the positions and derivatives are known, it is possible to solve for all coefficients of each polynomial analytically using the boundary conditions:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \Delta s^7 & \Delta s^6 & \Delta s^5 & \Delta s^4 & \Delta s^3 & \Delta s^2 & \Delta s & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 7\Delta s^6 & 6\Delta s^5 & 5\Delta s^4 & 4\Delta s^3 & 3\Delta s^2 & 2\Delta s & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 42\Delta s^5 & 30\Delta s^4 & 20\Delta s^3 & 12\Delta s^2 & 6\Delta s & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 210\Delta s^4 & 120\Delta s^3 & 60\Delta s^2 & 24\Delta s & 6 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A}_n \\ \mathbf{B}_n \\ \mathbf{C}_n \\ \mathbf{D}_n \\ \mathbf{E}_n \\ \mathbf{F}_n \\ \mathbf{G}_n \\ \mathbf{H}_n \end{bmatrix} = \begin{bmatrix} \mathbf{q}[n] \\ \mathbf{q}[n+1] \\ \frac{d\mathbf{q}[n]}{ds} \\ \frac{d\mathbf{q}[n+1]}{ds} \\ \frac{d^2\mathbf{q}[n]}{ds^2} \\ \frac{d^2\mathbf{q}[n+1]}{ds^2} \\ \frac{d^3\mathbf{q}[n]}{ds^3} \\ \frac{d^3\mathbf{q}[n+1]}{ds^3} \end{bmatrix} \tag{3.40}$$

which has a trivial solution. The result is a piecewise polynomial function, $\mathbf{q}(s)$ that connects the joint positions generated by Algorithm 1 in a jerk continuous way. By substituting $s[k]$ into



$\begin{bmatrix} \mathbf{Q} \\ \mathbf{s} \end{bmatrix}$
Quintic Spline Fitting
Feed Optimization
$C(s)$
$\dot{s}[mm/s]$
$\dot{s}(s)$
$s[mm]$
$\dot{s}(s)$
Taylor Approximation
$s[k+1] = s[k] + \dot{s}(s[k])T_s + \frac{\ddot{s}(s[k])}{2}T_s^2$
$s[k]$
$\boldsymbol{q}(s)$
$\begin{bmatrix} x_c[k] \\ y_c[k] \\ z_c[k] \\ x_f[k] \\ y_f[k] \\ z_f[k] \\ a_f[k] \\ b_f[k] \\ c_f[k] \end{bmatrix}$
To Machine

Figure 3.9: Real-time trajectory generation command from the result of feed optimization algorithm and 9-axis kinematic module

the $\mathbf{q}(s)$, it is possible to get $\mathbf{q}[k]$, which results in axes position commands for each of the axes at the controller sample period as illustrated in Fig. 3.9.

Chapter 4

Control Design for Micromill Feed Drives, High Precision Rotary Table, and Dual Stage Feed Drives

4.1 Overview

On the presented configuration there are 9 individual axes to control, with 3 from the micromill and 6 from the rotary table. Due to their differing dynamic characteristics, different approaches are used to identify and control the different systems. For the micromill, a sliding mode controller with a LuGre friction compensator are designed for each of the axes. It was found that each of the axes was able to achieve a bandwidth of about 30Hz before there were issues of stability. For the rotary table, linear lead-lag controllers were designed to control the positions. In order to push the bandwidth higher, notch filters were used to filter out resonant frequencies coming from the flexibilities of the rotary table setup. With the lead lag controllers and notch filters, the axes of the rotary table were able to have bandwidths of about 250Hz. Since the rotary table is able to achieve approximately 10 times the bandwidth of the micromill, and the translational degrees of freedom are redundant with the micromill, the rotary table is used to compensate for the tracking errors of the micromill. As a result, certain axes on the 9-axis micromachining center are able to have the long stroke of the micromill axes with the high bandwidth of the rotary table.

4.2 Modeling and Control Design of the Micromill

4.2.1 Identification of Rigid Body Dynamics

In order to design a control law for the axes of the micromill, a model for the feed drive must be developed and its corresponding parameters must be found. The approach outlined in [54] is adapted for the configuration in this work. In this methodology, the inertia, viscous friction, and Coulomb friction are found using linear regression. Based on these parameters, a pole placement controller and a Kalman Filter, as outlined in Section 4.2.2, is designed to identify non-linear friction characteristics. Using the x-axis as an example, the micromill can be modeled in frequency domain as a rigid body:

$$x(s) = \frac{K_a K_t}{ms^2 + bs} \left(u(s) - \frac{F_d(s)}{K_a K_t} \right) \tag{4.1}$$

where K_a is the amplifier gain, K_t is the motor gain, m is the mass of the feed drive, b is the viscous friction, $u(s)$ is the input signal, and $F_d(s)$ is the disturbance force. By considering the disturbance force as an equivalent disturbance input:

$$d(s) = \frac{F_d(s)}{K_a K_t} \tag{4.2}$$

and defining the gain $K_v = K_t K_a / m$ and the pole $p_v = -b/m$, the feed drive velocity, $v(s)$ can be expressed in terms of the control input, $u(s)$, and disturbance input, $d(s)$:

$$v(s) = \frac{K_v}{s - p_v} \left[u(s) - d(s) \right] \tag{4.3}$$

Since the feed drive is controlled with a digital real-time controller, the controller sees the plant defined in Eq.(4.3) as a plant discretized by the zero order hold:

$$v(k) = \frac{K_{vd}}{z - p_{vd}} \left[u(k) - d(k) \right] \tag{4.4}$$

where

$$K_{vd} = \frac{K_v}{-p_v}(1 - e^{p_v T_s}) \tag{4.5}$$

$$p_{vd} = e^{p_v T_s} \tag{4.6}$$

From the discrete transfer function Eq.(4.4), the following difference equation can be found:

$$v(k) = p_{vd} v(k-1) + K_{vd} u(k-1) - K_{vd} d(k-1) \tag{4.7}$$

In the case of a linear feed drive, and in the absence of process forces, the major source of disturbance comes from velocity dependent Coulomb friction which can be modeled as:

$$F_f(v) = \begin{cases} 0 & \text{if } v = 0 \\ F^+_{\text{coul}} & \text{if } v > 0 \\ F^-_{\text{coul}} & \text{if } v < 0 \end{cases} \tag{4.8}$$

which must be modeled as an equivalent disturbance and included into the difference equation as $d(k)$. The equivalent disturbances, corresponding to the Coulomb friction, is found as follows:

$$d_f^+ = \frac{F^+_{\text{coul}}}{K_a K_t} \tag{4.9}$$

$$d_f^- = \frac{F^-_{\text{coul}}}{K_a K_t} \tag{4.10}$$

Based on the friction model in Eq.(4.8) and the equivalent disturbances in Eq.(4.9) and Eq.(4.10), the disturbance from the Coulomb friction is defined as follows:

$$d_f(v(k)) = PV(v(k)) \cdot d_f^+ + NV(v(k)) \cdot d_f^- \tag{4.11}$$

where $PV(v(k))$ and $NV(v(k))$ are the positive and negative velocity functions defined as follows:

$$PV(v(k)) = \frac{1}{2} \cdot \sigma\left(v(k)\right) \cdot \left(1 + \sigma\left(v(k)\right)\right) \tag{4.12}$$

$$NV(v(k)) = -\frac{1}{2} \cdot \sigma\left(v(k)\right) \cdot \left(1 - \sigma\left(v(k)\right)\right) \tag{4.13}$$

and where σ is a deadband function to ensure encoder noise does not affect the identification:

$$\sigma\left(v, V_d\right) = \begin{cases} 0 & \text{if } |v| \leq V_d \\ 1 & \text{if } v > V_d \\ -1 & \text{if } v < -V_d \end{cases} \tag{4.14}$$

where V_d is the limit in which the measured velocity is considered to be non-zero. This ensures that noise from the velocity measurement is not mistaken for actual movement, which may potentially affect the identification results. With the equivalent disturbance defined in Eq. (4.11), the difference equation can be rewritten as:

$$v(k) = p_{vd}v(k-1) + K_{vd}u(k-1) - \left[K_{vd}d_f^+ PV(v(k-1)) + K_{vd}d_f^- NV(v(k-1))\right] \tag{4.15}$$

A frequency rich signal, u, as shown in Fig. 4.1 is sent to the machine and the velocity is measured. This type of signal is selected in order to excite the system at as many frequencies as possible, resulting in better fitting results. Since the velocity, v, is measured by the encoder, it also defines the speed dependent sign functions, $PV(v)$ and $NV(v)$. The only unknowns in Eq. (4.15) are the coefficients p_{vd}, K_{vd}, d_f^+, and d_f^-. The problem can be redefined as a matrix

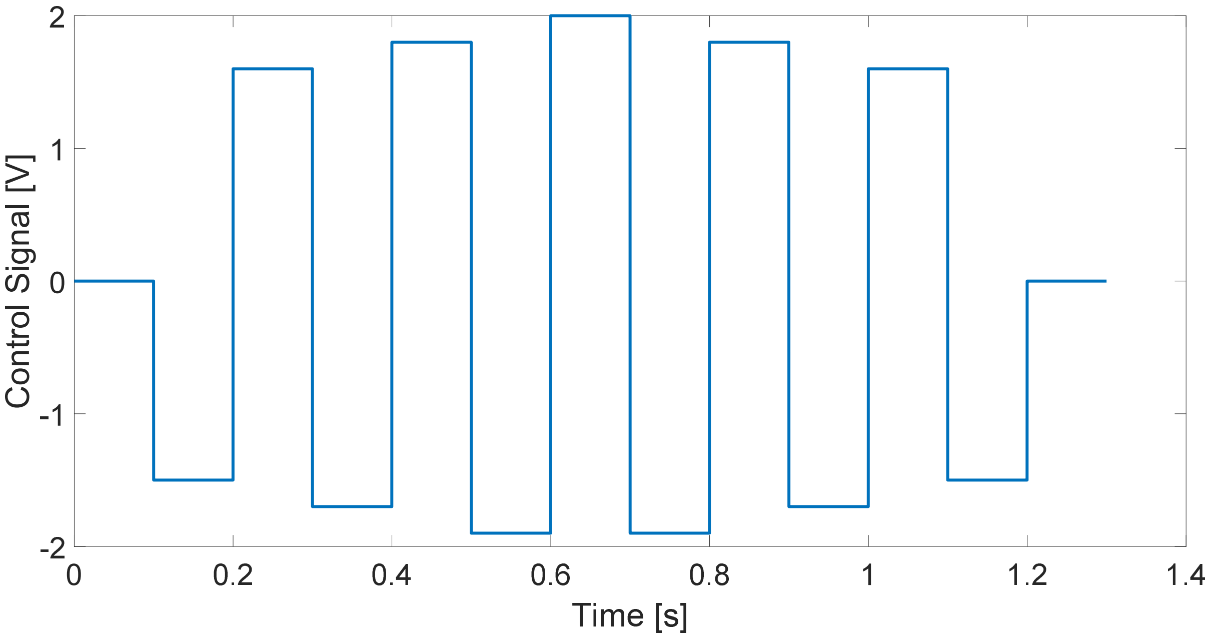


Figure 4.1: Frequency rich signal for rigid body identification

problem as follows:

$$\underbrace{\begin{bmatrix} v(2) \\ v(3) \\ \vdots \\ v(N) \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} v(1) & u(1) & PV(1) & NV(1) \\ v(2) & u(2) & PV(2) & NV(2) \\ \vdots & \vdots & \vdots & \vdots \\ v(N-1) & u(N-1) & PV(N-1) & NV(N-1) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} p_{vd} \\ K_{vd} \\ K_{vd}d_f^+ \\ K_{vd}d_f^- \end{bmatrix}}_{\theta} \tag{4.16}$$

where N is the total number of measurements in the identification experiments. The objective is to find coefficients in θ that model the relationship as accurately as possible. Since $\mathbf{Y}$ is known, and Φ is known, the coefficients in θ are found in order to minimize the error between the predicted and measured value, which can be phrased in the following minimization problem:

$$\min \frac{1}{2}(\mathbf{Y} - \Phi\theta)^T(\mathbf{Y} - \Phi\theta) \tag{4.17}$$

Table 4.1: Identified rigid body parameters of the micromill

Parameter	x	y	z
$\hat{m}$ [kg]	21.19	59.17	17.61
$\hat{b}$ [kg/s]	325.91	346.01	179.10
$\hat{F}_{\text{coul}}^{+}$ [N]	6.29	19.98	17.47
$\hat{F}_{\text{coul}}^{-}$ [N]	-6.094	-18.93	-18.28

which can be minimized by setting θ as follows:

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{Y} \tag{4.18}$$

From θ the estimated values for mass ($\hat{m}$), viscous friction ($\hat{b}$), and coulomb friction ($\hat{F}_{\text{coul}}^{+/-}$) can be found as follows:

$$\hat{m} = \frac{(\hat{p}_{vd} - 1) K_t K_a T_s}{\hat{K_{vd}} \ln(\hat{p}_{vd})} \tag{4.19}$$

$$\hat{b} = \frac{(1 - \hat{p}_{vd}) K_t K_a}{\hat{K_{vd}}} \tag{4.20}$$

$$\hat{F}_{\text{coul}}^{+} = \frac{\hat{K_{vd} d_f^{+}}}{\hat{K_{vd}}} \tag{4.21}$$

$$\hat{F}_{\text{coul}}^{-} = \frac{\hat{K_{vd} d_f^{-}}}{\hat{K_{vd}}} \tag{4.22}$$

From these tests, the corresponding mass, viscous friction, and coulomb friction values for each axis are found and outlined in Tab. 4.1. Based upon these results the friction model is improved as outlined in the following section.

4.2.2 Identification of Friction Characteristics

In addition to the Coulomb friction modeled and identified in Section 4.2.1, feed drives have additional non-linear friction phenomenon, which must be identified and compensated in order to improve tracking performance. To identify this non-linear friction characteristic, a Kalman filter is designed to observe the disturbance when the axis is jogged back and forth at various speeds. In the absence of process forces, the disturbances observed by the Kalman filter corresponds to non-linear friction phenomena. A simple pole placement controller is first designed, based on the rigid body parameters found in the previous section, allowing the axis to be jogged at different speeds. It should be noted that the pole placement controller is used for identification purposes, and a more sophisticated sliding mode controller is developed in the following section.

In order to design the Kalman filter, the feed drive must be modeled in discrete state space form, with disturbance as one of the states while considering the measurement and input noise. With the axis velocity defined in Eq. (4.3), the axis position can be found as follows:

$$x(s) = \frac{1}{s}\frac{K_v}{s - p_v}\left[u(s) - d(s)\right] \tag{4.23}$$

which gives the following state space model:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}(t) \end{bmatrix} = \mathbf{A}_c \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \mathbf{B}_c \begin{bmatrix} u(t) \\ d(t) \end{bmatrix} \tag{4.24}$$

where $\mathbf{A}_c$ and $\mathbf{B}_c$ are the transition and input matrices defined as:

$$\mathbf{A}_c = \begin{bmatrix} 0 & 1 \\ 0 & p_v \end{bmatrix} \tag{4.25}$$

$$\mathbf{B}_c = \begin{bmatrix} 0 & 0 \\ K_v & -K_v \end{bmatrix} \tag{4.26}$$

From this state space model, the discrete time state is found:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \mathbf{A}_d \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \mathbf{B}_d \begin{bmatrix} u(k) \\ d(k) \end{bmatrix} \tag{4.27}$$

where $\mathbf{A}_d$ and $\mathbf{B}_d$ are the discrete transition and input matrices defined as:

$$\mathbf{A}_d = e^{\mathbf{A}_c T_s} \tag{4.28}$$

$$\mathbf{B}_d = \int_0^{T_s} e^{\mathbf{A}_c \lambda} d\lambda \cdot \mathbf{B}_c \tag{4.29}$$

In order to design a Kalman filter that can estimate disturbances in the presence of noise, the discrete state space model in Eq.(4.27) is augmented in the following way:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \\ d(k+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} x(k) \\ v(k) \\ d(k) \end{bmatrix} + \mathbf{B} \begin{bmatrix} u(k) \end{bmatrix} + \mathbf{W} \begin{bmatrix} \tilde{u}(k) \\ w_d(k) \end{bmatrix} \tag{4.30}$$

$$\begin{bmatrix} x_m(k) \\ v_m(k) \end{bmatrix} = \mathbf{C} \begin{bmatrix} x(k) \\ v(k) \\ d(k) \end{bmatrix} + \mathbf{V} \begin{bmatrix} \tilde{x}(k) \\ \tilde{v}(k) \end{bmatrix} \tag{4.31}$$

where $\tilde{u}(k)$ is the input noise, $\tilde{x}(k)$ is the feedback position noise, $\tilde{v}(k)$ is the velocity feedback noise, and $w_d(k)$ is the disturbance noise. The matrices in the augmented state space model are defined as:

$$\mathbf{A} = \begin{bmatrix} \multicolumn{2}{c}{\mathbf{A}_d} & -\mathbf{B}_d \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_d \\ 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} \mathbf{B}_d & \begin{matrix} 0 \\ 0 \end{matrix} \\ 0 & 1 \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.32}$$

The augmented state space model treats the disturbance as a state, allowing the Kalman filter to observe it. Next, the noise of the feedback and input must be considered in order to select optimal gains for the Kalman filter. The position of the feed drives are measured with a linear encoder with a resolution of δx = 40 [nm], which introduces an error in measurement due to quantization. As a result, the error between the actual and measured position is uniformly distributed with a zero mean and values between $+\delta x/2$ and $-\delta x/2$. The variance can be computed as:

$$R_{\tilde{x}} = E[(\tilde{x} - E[\tilde{x}]) = E[\tilde{x}^2] = \int_{-\infty}^{\infty} p\left(\tilde{x}\right) \cdot \tilde{x}^2 d\tilde{x} = \int_{-\delta x/2}^{\delta x/2} \tilde{x}^2 d\tilde{x} = \frac{\left(\delta x\right)^2}{12} \tag{4.33}$$

Since the noise is the digitally differentiated encoder signal, then the resolution is δv = 40/T_s [nm/s] resulting in a variance of:

$$R_{\tilde{v}} = \frac{\left(\delta v\right)^2}{12} = \frac{\left(\delta x/T_s\right)^2}{12} \tag{4.34}$$

Similar to the feedback, quantization of the input signal can introduce additional noise into the system. The current amplifier is controlled with a PWM control signal with a frequency of 50kHz. The real-time controller of dSpace is capable of varying the the PWM with a resolution of 50ns. As a result, within a duty cycle, it is possible to have 400 unique values. Since the amplifier has a saturation limit of $\pm 5[V]$ then the resolution of the input signal is δu = 10/400[V] = 0.025[V], which, similar to the feedback, gives a variance of:

$$R_{\tilde{u}} = \frac{\left(\delta u\right)^2}{12} \tag{4.35}$$

Unlike the covariances $R_{\tilde{x}}$, $R_{\tilde{v}}$, and $R_{\tilde{u}}$, R_{w_d} is a tuning parameter that adjusts the performance of the Kalman filter with respect to predicting disturbance. A higher covariance results in a Kalman filter that converges faster but outputs noisier results and vice versa for lower covariance. It was found the following values resulted in the best balance between performance and noise:

$$R_{w_d} = 1 \times 10^{-6} \tag{4.36}$$

Given these parameters, the Kalman filter takes the form:

$$\begin{bmatrix} \hat{x}(k) \\ \hat{v}(k) \\ \hat{d}(k) \end{bmatrix} = (\mathbf{I} - \mathbf{K}_{\text{obs}}\mathbf{C})\,\mathbf{A} \begin{bmatrix} \hat{x}(k-1) \\ \hat{v}(k-1) \\ \hat{d}(k-1) \end{bmatrix} = (\mathbf{I} - \mathbf{K}_{\text{obs}}\mathbf{C})\,\mathbf{B} \begin{bmatrix} u(k-1) \end{bmatrix} + \mathbf{K}_{\text{obs}} \begin{bmatrix} x_m(k) \\ v_m(k) \end{bmatrix} \tag{4.37}$$

where $\hat{x}(k)$, $\hat{v}(k)$, and $\hat{d}(k)$ are the estimated position, velocity, and disturbance, respectively, at time step k, $\mathbf{K}_{\text{obs}}$ is the Kalman filter gain matrix and $x_m(k)$ and $v_m(k)$ are the measured axis position and velocity, respectively. Based on these system parameters and noise variances, the Kalman observer gain matrix, $\mathbf{K}_{\text{obs}}$, is found [55] [19], giving us the following observer gains for the x, y, and z axis as follows:

$$\mathbf{K}_{\text{obs},x} = \begin{bmatrix} 0.2227 & 2.8518\text{E}-6 \\ 285.1827 & 0.0061 \\ -73.5388 & -0.0020 \end{bmatrix} \tag{4.38}$$

$$\mathbf{K}_{\text{obs},y} = \begin{bmatrix} 0.1880 & 1.9796\text{E}-6 \\ 197.9680 & 0.0034 \\ -76.1335 & -0.0017 \end{bmatrix} \tag{4.39}$$

$$\mathbf{K}_{\text{obs},z} = \begin{bmatrix} 0.1993 & 2.2430\text{E}-6 \\ 224.3086 & 0.0042 \\ -75.3177 & -0.0018 \end{bmatrix} \tag{4.40}$$

With these Kalman filters, the axes are jogged back and forth at ±1 [mm/s], ±3 [mm/s], ±5 [mm/s], ±10 [mm/s], ±20 [mm/s], ±30 [mm/s], ±40 [mm/s], ±50 [mm/s], ±75 [mm/s], ±100 [mm/s], ±125 [mm/s], and ±150 [mm/s] and the disturbance is measured over the whole stroke of the feed drive. An example of the disturbance prediction at 10 [mm/s] can be seen in Fig. 4.2. The disturbance over the whole stroke of the feed drive is averaged, resulting in a disturbance estimate at each corresponding velocity as shown in Fig. 4.3.

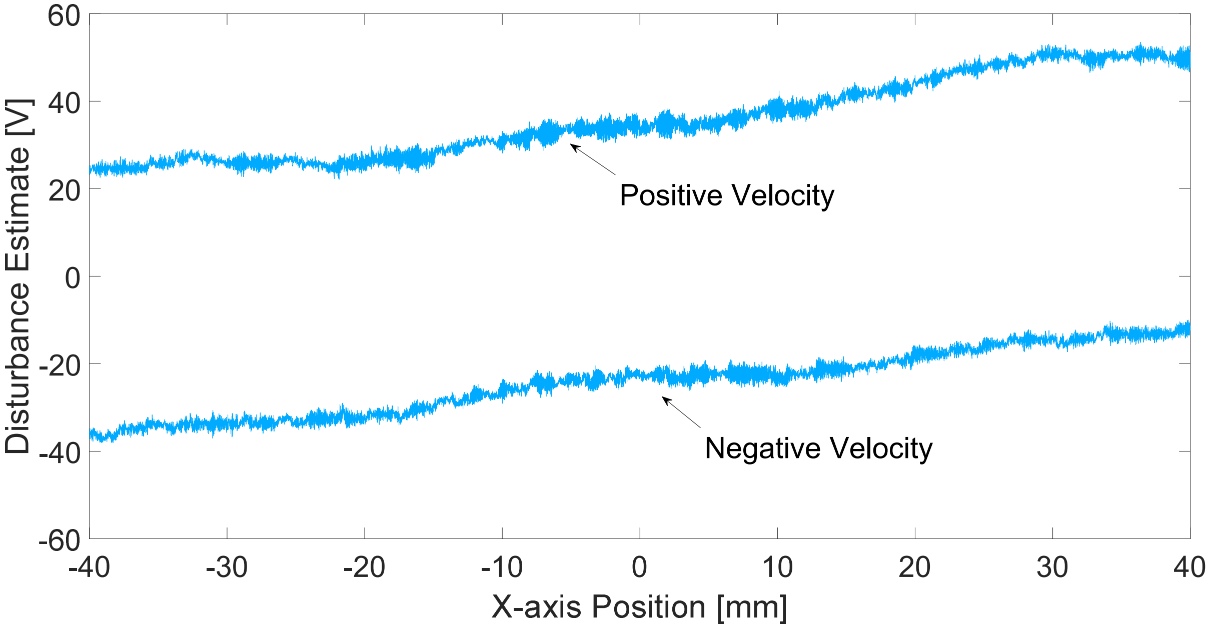


Figure 4.2: Disturbance estimation for x_c-axis at 10 [mm/s]

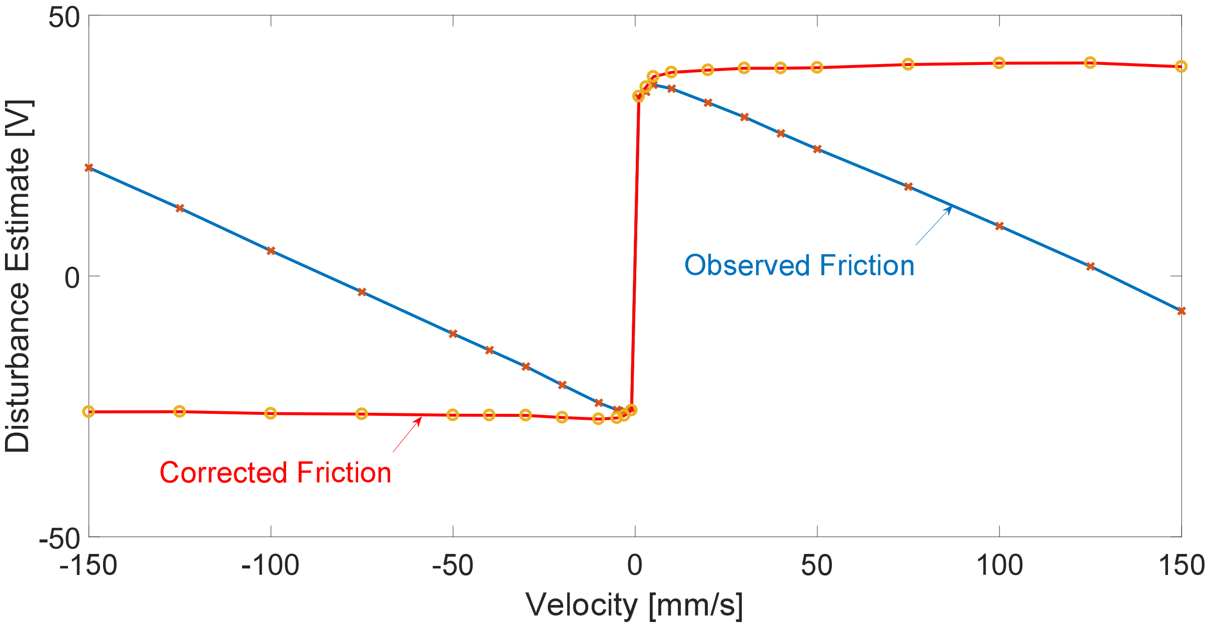


Figure 4.3: Average disturbance estimation for x_c-axis at all test speeds

From the velocity based disturbance data a Stribeck friction model of the form:

$$F_f(v) = \begin{cases} F^{+}_{\text{stat}} e^{-v/\Omega^{+}_{1}} + F^{+}_{\text{coul}} \left(1 - e^{-v/\Omega^{+}_{2}}\right) + bv, & \text{for } v > 0 \\ F^{-}_{\text{stat}} e^{-v/\Omega^{-}_{1}} + F^{-}_{\text{coul}} \left(1 - e^{-v/\Omega^{-}_{2}}\right) + bv, & \text{for } v < 0 \end{cases} \tag{4.41}$$

is fit. In Eq.(4.41) F^{+}_{stat} and F^{-}_{stat} are the static frictions in the positive and negative directions respectively, F^{+}_{coul} and F^{-}_{coul} are the Coulomb frictions in the positive and negative directions respectively, and Ω^{+}_{1}, Ω^{-}_{1}, Ω^{+}_{2}, and Ω^{-}_{2}, are exponential coefficients which determine the rate in which the static friction converges to the Coulomb friction.

Since the Kalman observer is based on the parameters identified in the rigid body model the disturbance should consist of only the non-linear friction phenomenon. However, it is possible that the viscous friction, b, from the openloop rigid body identification may be inaccurate. Should this be the case, the disturbance will exhibit linear variation at the higher speeds, where the disturbance should only be defined by a constant Coulomb friction. An example of this can be seen in Fig. 4.3, where the observed friction has a negative slope at higher velocities, indicating a viscous friction that is higher than one obtained from the identification. If this is the case, the originally identified viscous friction coefficient is adjusted by first identifying the slopes in the disturbance estimate in the positive, $\triangle b^{+}$, and negative direction, $\triangle b^{-}$, using linear regression. An average slope is found:

$$\triangle \bar{b} = \frac{\triangle b^{+} + \triangle b^{-}}{2} \tag{4.42}$$

and the original viscous friction is updated as follows:

$$\hat{b}' = \hat{b} + \triangle \bar{b} \tag{4.43}$$

From Fig. 4.3, it can be seen that the slopes at higher velocities are removed once the viscous friction has been corrected. Once corrected, the viscous friction in the Kalman filter is updated

Table 4.2: Identified friction parameters of the micromill

Parameter	x	y	z
b [kg/s]	25.05	20.55	102.9
F_{stat}^{+} [N]	36.38	43.2902	30.6853
F_{stat}^{-} [N]	-29.03	-29.2014	-33.9802
F_{coul}^{+} [N]	39.74	47.9513	23.6214
F_{coul}^{-} [N]	-26.74	-35.5574	-23.5320
Ω_1^{+} [mm/s]	1.31	8.81	6.2
Ω_1^{-} [mm/s]	-0.34	-0.74	-6.8
Ω_2^{+} [mm/s]	1.79	10	4.2
Ω_2^{-} [mm/s]	-0.45	-0.74	-4.1

and the velocity tests are ran so the non-linear friction parameters can be identified more accurately. The disturbance with the corrected viscous friction can be seen in Fig. 4.4. In order to fit the parameters in the Stribeck model, F_{stat}^{+} and F_{stat}^{-} are first selected as the disturbance force at the lowest velocities in the positive and negative direction, respectively. Since the non-linear friction affects associated with static friction go to zero at high velocities, the Coulomb friction, F_{coul}^{+} and F_{coul}^{-}, is selected as the average disturbance force of the four highest velocities. Finally, the exponential coefficients are found by minimizing the cost function:

$$V = \frac{1}{2} \sum_{k=N_1}^{N_2} [\delta F(k)]^2 \tag{4.44}$$

where $\delta F(k)$ is the difference between the measured disturbance and predicted disturbance:

$$\delta F(k) = \delta F[v(k)] = F_f'[v(k)] - \hat{F}_{\text{stat}}^{+/-} e^{-v/\Omega_1^{+/-}} + \hat{F}_{\text{coul}}^{+/-} \left(1 - e^{-v/\Omega_2^{+/-}}\right) \tag{4.45}$$

Due to the non-linear and discrete nature of Eq. (4.45) the cost function is minimized in a brute force way. All permutations between 0 and 10 for are tested at increments of 0.1 [mm/s]. Based upon this friction identification methodology, the parameters shown in Tab. 4.2 are found.

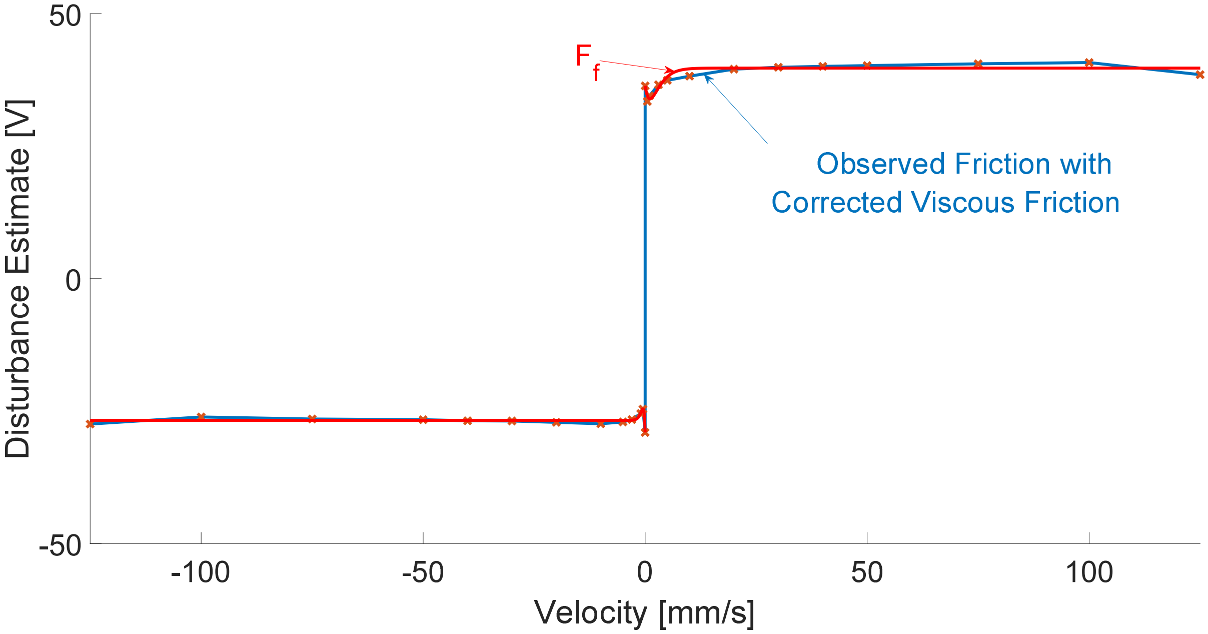


Figure 4.4: Disturbance estimation for x_c-axis with corrected viscous friction and the fitted friction curve

With the above coefficients, the friction behaviour, as described by the model, can be seen in Fig. 4.4. Using the identified mass, corrected viscous friction, and non-linear friction parameters, a sliding mode controller and a feedfoward friction compensator, based on the LuGre model, is developed.

4.2.3 Sliding Mode Position Controller

Unlike conventional ballscrews, where the pitch of the ball screw reduces the reflected disturbance on the motor, a linear feed drive experiences a disturbance force completely. As a result, a controller must be selected which has good disturbance rejection properties. Furthermore, specific to the micromill in this work, the friction properties have significant variation over the length of the travel. As a result, the sliding mode controller [56] was selected as the position controller for the feed drives of the micromill. The design of the controller is based on work presented in [21] but has been extended to linear feed drives. From Eq. (4.1) the corresponding time domain state space model can be found:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -b_e/m_e \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m_e \end{bmatrix} \cdot u - \begin{bmatrix} 0 \\ 1/m_e \end{bmatrix} \cdot d \tag{4.46}$$

where $m_e = m/(K_a K_t)$ and $b_e = b/(K_a K_t)$ and the indication for function of time (t) is dropped for brevity. Since the goal of the feed drive is to follow a trajectory, a sliding surface is designed to bring the tracking error to zero:

$$\sigma = (\dot{x}_r - \dot{x}) + \lambda\,(x_r - x) = \underbrace{[\lambda \; 1]}_{S} \cdot \left(\begin{bmatrix} x_r \\ \dot{x}_r \end{bmatrix} - \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \right) = 0 \tag{4.47}$$

where λ is analogous to the bandwidth of the sliding mode controller, x and $\dot{x}$ are the position and velocity of the axis, respectively, and x_r and $\dot{x}_r$ are the reference position and reference velocity of the axis respectively. The objective in sliding mode control is to bring the system onto the sliding surface $\sigma = 0$, which then brings the tracking error to zero. In order to design a control law that drives the system to the sliding surface and is stable, a candidate Lyapunov function is first selected as a function of the sliding surface and a disturbance estimator as follows:

$$V(t) = \frac{1}{2} m_e \sigma^2 + \frac{\left(d - \hat{d}\right)^2}{2\rho} \tag{4.48}$$

where d is the actual disturbance, $\hat{d}$ is the disturbance estimator, and ρ is the adaption gain of the disturbance estimator. The disturbance estimator acts similar to an integrator and is defined as a function of the sliding surface:

$$\hat{d}[k+1] = \hat{d}[k] + \rho\kappa\sigma T_s \tag{4.49}$$

T_s is the sample time, and κ is a flag to ensure the disturbance estimator does not go beyond the predetermined disturbance bound as follows:

$$\kappa = \left\{ \begin{array}{lr} 0, & \text{if } \hat{d} \leq d^{-} \text{ and } \sigma \leq 0 \\ 0, & \text{if } \hat{d} \geq d^{+} \text{ and } \sigma \geq 0 \\ 1, & \text{otherwise} \end{array} \right\} \tag{4.50}$$

As it can be seen the Lyapunov function is directly proportional to the size of sliding surface and the error in disturbance estimation. If the control law is selected so that the time derivative of the Lyapunov function is less than zero, the sliding surface and error in disturbance estimation will be guaranteed to converge to zero. In order to do this, we take the time derivative of Eq. (4.48) and set the derivative to be less than 0:

$$\frac{dV(t)}{dt} = m_e \sigma \dot{\sigma} + \frac{\left(d - \hat{d}\right)}{\rho} \dot{\hat{d}} < 0 \tag{4.51}$$

The derivative of the sliding surface and disturbance estimate is found as follows:

$$\dot{\sigma} = \underbrace{[\lambda \ 1]}_{S} \cdot \left(\begin{bmatrix} \dot{x}_r \\ \ddot{x}_r \end{bmatrix} - \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} \right) \tag{4.52}$$

$$\dot{\hat{d}} = \rho \kappa \sigma \tag{4.53}$$

Substituting Eq. (4.46) into Eq. (4.52) into , the derivative of the sliding mode can be redefined as a function of the rigid body model:

$$\dot{\sigma} = \underbrace{[\lambda \ 1]}_{S} \cdot \left(\begin{bmatrix} \dot{x}_r \\ \ddot{x}_r \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 0 & -b_e/m_e \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m_e \end{bmatrix} \cdot u - \begin{bmatrix} 0 \\ 1/m_e \end{bmatrix} \cdot d \right) \tag{4.54}$$

Then by substituting the sliding surface (Eq. (4.47)), the disturbance estimate (Eq. 4.49), and their respective derivatives (Eq.(4.54) and Eq. (4.53)) into Eq. (4.51), the derivative of the Lyapunov equation can be redefined as a function of known system parameters:

$$\frac{dV(t)}{dt} = m_e \sigma (\lambda \left(\dot{x}_r - \dot{x}\right) + \ddot{x}_r) - \sigma \left(u - d - b_e \dot{x}\right) - \sigma \kappa (d - \hat{d}) \tag{4.55}$$

If Eq. (4.55) is always less than 0 then the sliding mode controller will be asymptotically stable. In Eq. (4.55), $\sigma d - \sigma\kappa(d - \hat{d})$ can be set to $\sigma\hat{d} + \sigma(d - \hat{d})(1 - \kappa)$. Given the bounds set in Eq. (4.50), then the condition $\sigma(d - \hat{d})(1 - \kappa) \leq 0$ is always guaranteed. To ensure asymptotic stability for the remainder of the Lyapunov equation, the following criteria is set:

$$m_e\sigma(\lambda\left(\dot{x}_r - \dot{x}\right) + \ddot{x}_r) - \sigma\left(u - b_e\dot{x} - \hat{d}\right) = -K_s\sigma^2 \tag{4.56}$$

Since Eq. (4.56) is the criteria for asymptotic stability of the sliding mode controller, the control signal u can be found that corresponds to this criteria. In order to do this Eq. (4.56) is rearranged to find u as follows:

$$u_{smc} = m_e\left[\lambda\left(\dot{x}_r - \dot{x}\right) + \ddot{x}_r\right] + b_e\dot{x} + \hat{d} + K_s\sigma \tag{4.57}$$

which ensures that the system will converge to the sliding surface, and subsequently bring tracking error to 0.

4.2.4 LuGre Feedforward Friction Compensator

Though the sliding mode controller is capable of rejecting disturbance, it is only capable of doing so after some error has been incurred. Due to the non-linear friction phenomenon, at points where the velocity switches signs a disturbance occurs which causes a tracking error. Since this non-linear friction phenomenon can be modeled with relative accuracy, it is possible to compensate for its effects before the occur. In this work, the LuGre friction model [57] is used to model the non-linear friction effects. The LuGre friction model is an extension of the traditional Stribeck friction model, as defined in Eq. (4.41), which is able to capture the hysteretic effects of friction. Unlike the Coulomb or Stribeck friction model, the LuGre friction model does not treat the non-linear friction phenomenon as discontinuous force changes at points of velocity switching. Instead, it captures the friction dynamics in the stiction region,

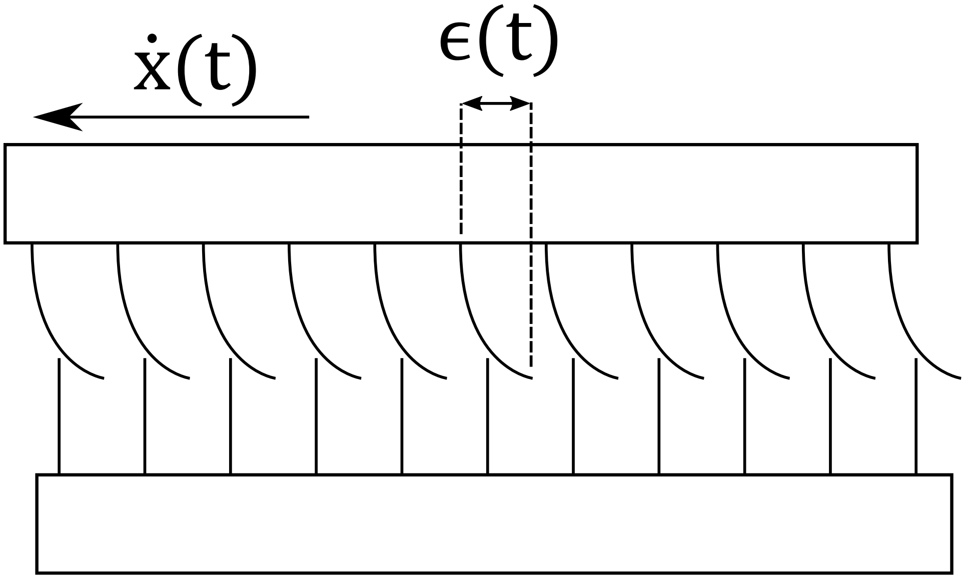


Figure 4.5: Graphical representation of LuGre friction model

which come from the deflection, $\epsilon(t)$, of micro structures between surfaces, before the static, Coulomb, and viscous friction affect become dominant. The microstructures are analogous to bristles and $\epsilon(t)$ can be seen as the average deflections of these bristles as shown in Fig.4.5.

Based upon this model, the deflections of $\epsilon(t)$ are governed with the following differential equation:

$$\frac{d\epsilon(t)}{dt} = \dot{x}(t) - \frac{\sigma_0}{g(\dot{x}(t))}\epsilon(t)\left|\dot{x}(t)\right| \tag{4.58}$$

where σ_0 is analogous to the stiffness constant for the microstructures, which is found experimentally [58], and the speed dependent function, $g(\dot{x}(t))$ is expressed as:

$$g(\dot{x}(t)) = F_c + F_s \mathrm{e}^{\frac{\dot{x}(t)}{\Omega_s}} \tag{4.59}$$

where F_C is coulomb friction, F_S is the velocity ($\dot{x}(t)$) dependent static friction, and Ω_s is the velocity in which the effect of static friction decays to 0. Equation (4.59) can be seen as a simplified, averaged of Eq. (4.41), and as a result, the Stribeck friction effects are incorporated into the LuGre model through Eq. (4.59). The constants are found using the results of the friction identification experiments in Section 4.2.2 as follows:

$$F_s = \frac{1}{2}\left(F^{+}_{\text{stat}} + |F^{-}_{\text{stat}}|\right) \tag{4.60}$$

$$F_c = \frac{1}{2}\left(F^{+}_{\text{coul}} + |F^{-}_{\text{coul}}|\right) \tag{4.61}$$

and Ω_s is found by minimizing Eq. (4.45) using Eq. (4.59). With the microstructure deflections modeled in Eq. (4.58), the instantaneous friction force, $F_f(t)$, can be modeled as follows:

$$F_f(t) = \sigma_0\epsilon(t) + \sigma_1\frac{d\epsilon(t)}{dt} + b\dot{x}(t) \tag{4.62}$$

where σ_1 is analogous to the damping constant [58] of the microstructures. The stiffness and damping of the microstructures, σ_0 and σ_1 respectively, are first approximated with the approach outlined in [58]. For σ_0, a very small step x_{ss}, assumed to be in the stiction region, can be given to the feed drive and the constant can be approximated as:

$$\sigma_0 = F_c\text{sgn}(\dot{x}(t))/x_{ss} \tag{4.63}$$

In the stiction region, the dynamics of the system can be modeled as:

$$m\ddot{x} + (b + \sigma_1 + \sigma_2)\,\dot{x} + \sigma_0 x = u(t) \tag{4.64}$$

and σ_1 can be calculated as:

$$\sigma_1 = 2\sqrt{m\sigma_0} - b - \sigma_2 \tag{4.65}$$

With the constants defined in Eq. (4.63) and Eq. (4.65), the values were tuned in tracking experiments to achieve the best performance. Since the sliding mode controller is able to suppress errors contributed by viscous friction, the feed forward friction compensator only has to compensate for the forces contributed by static friction, Coulomb friction, and microstructure deflection $\epsilon(t)$. As a result, the feed forward compensator has the following structure:

$$u_{ff} = \frac{1}{K_a K_t}\left(\sigma_0 \epsilon(t) + \sigma_1 \frac{d\epsilon(t)}{dt}\right) \tag{4.66}$$

which represents the LuGre friction modeled with the viscous friction removed and normalized by the axis amplifier and motor gain. The deflection $\epsilon(t)$ is found by numerically solving Eq. 4.58 using the Forward Euler Approximation:

$$\begin{aligned}
&\dot{x}[0] = 0 \quad \epsilon[0] = 0 \quad \frac{d\epsilon[0]}{dt} = 0 \\
&\frac{d\epsilon[t]}{dt} = \dot{x}[k] - \frac{\sigma_0}{g(\dot{x}[k])}\epsilon[k-1]\left|\dot{x}[k]\right| \\
&\epsilon[k] = \epsilon[k-1] + \frac{d\epsilon[k]}{dt}T_s
\end{aligned} \tag{4.67}$$

The full control law is the sum of sliding mode (Eq. 4.57) and feed forward compensator (Eq. 4.66) commands:

$$\begin{aligned}
u(k) &= u_{smc}(k) + u_{ff}(k) \\
&= m_e\left[\lambda\left(\dot{x}_r(k) - \dot{x}(k)\right) + \ddot{x}_r(k)\right] + b_e\dot{x}(k) + \hat{d}(k) \\
&\quad + K_s\sigma(k) + \tfrac{1}{K_a K_t}\left[\sigma_0\epsilon(k) + \sigma_1 d\epsilon(k)\right]
\end{aligned} \tag{4.68}$$

The block diagram of the full micromill control law is shown in Fig. 4.6. With the tuned LuGre friction parameters, the sliding mode controller is tuned to achieve as high a bandwidth as possible before instabilities occurred and the control parameters are shown in Tab. 4.3.

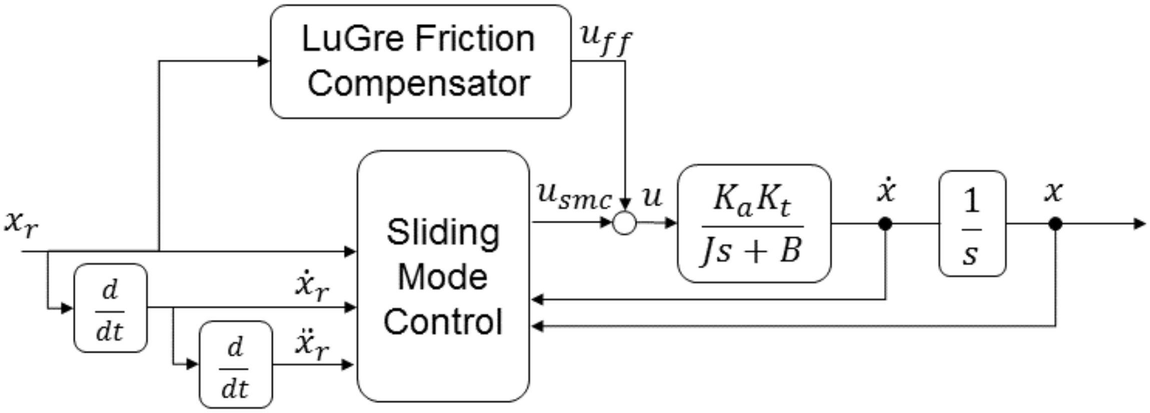


Figure 4.6: Block diagram of the sliding mode controller and LuGre friction compensator for the micromill

Table 4.3: Micromill parameters

Axis	x_c	y_c	z_c
K_a [A/V]	1	1	1
K_t [N/A]	45.4	72.5	26.17
K_s [Vs/mm]	0.05	0.25	0.08
λ [1/s]	200	230	180
ρ[V/mm]	50	50	50
σ_0 [N/mm]	900	900	1500
σ_1 [Ns/mm]	1.5	2.5	7.5
F_c [N]	26.67	31.26	24.67
F_s [N]	29.15	34.11	32.35
Ω_s [mm/s]	24.3	26.4	44.8

4.3 Modeling and Control Design of the Rotary Table

4.3.1 Lead-lag Position Controller

Unlike the feed drive, the rotary table is non-contact. As a result, the traditional rigid body model for linear feed drives can be simplified by eliminating the viscous friction. This eliminates the need for sophisticated identification techniques, as it is possible to obtain the mass of the rotary table by simply weighing it. It was found that the mass of the rotary table was 2.3kg and as a result the mass in the translational axes can be defined as:

$$J_x = J_y = J_z = 2.3[\text{kg}] \tag{4.69}$$

The rotational inertia about the x, y, and z axis were found based on the solid model of the rotary table [59]:

$$J_a = J_b = 20878[\text{kg} \cdot \text{mm}^2] \tag{4.70}$$

$$J_c = 40508[\text{kg} \cdot \text{mm}^2] \tag{4.71}$$

Since the rotary table is a free floating mass, the nominal transfer function for each axis of the rotary table can be modeled as follows:

$$G_f(s) = \frac{1}{J_q s^2} \leftarrow q \in [x, y, z, a, b, c] \tag{4.72}$$

Due to its relatively simple dynamics, lead-lag controllers were selected to control the position of the rotary table which take the form:

$$C_f(s) = K_f \frac{1 + \alpha T s}{1 + T s} \tag{4.73}$$

where α is selected to add a desired phase, T is selected to add the phase at a specific location, and K_f is selected so $C_f(s)G_f(s)$ crosses the 0dB line at a desired frequency. In order to do this, the parameters are calculated as follows:

$$\alpha = \frac{1+\sin\phi_m}{1-\sin\phi_m} \quad T = \frac{1}{\omega_m\sqrt{\alpha}} \quad K_f = \frac{1}{|\frac{1+\alpha T\omega_m j}{1+T\omega_m j}||G_f(\omega_m)|} \tag{4.74}$$

where ω_m is the desired crossover frequency of the openloop transfer function, which corresponds approximately to the bandwidth of the closed loop transfer function, and ϕ_m is the phase to be added. An integrator of the form:

$$C_I(s) = 1 + \frac{K_i}{s} \tag{4.75}$$

with $K_i = \omega_m/10$ is added to the control loop to reduce the steady state error. However, it was found that unlike the nominal models in Eq. (4.72), the rotary table was not completely rigid, hence its flexibility had to be considered while designing the controller. The resonance peaks caused by the flexibilities of the rotary table created unstable conditions when the design bandwidth was set to 250Hz. As a result the gain is attenuated at the frequency of the resonance peaks with notch filters so that the control action does not excite the low damped dynamics and its dynamic characteristics match the nominal model, as shown in Fig. 4.7.

4.3.2 Notch Filter

While designing the lead-lag controllers for the various axes of the rotary table, it was found that the nominal transfer function defined in Eq.(4.72) did not fully capture all the dynamics of the rotary table. At high enough frequencies, the rotary table was not completely rigid and exhibited flexible characteristics. An example of this is shown in Fig. 4.7 where the rigid nominal model is compared with the actual frequency sweep of the a_f-axis.

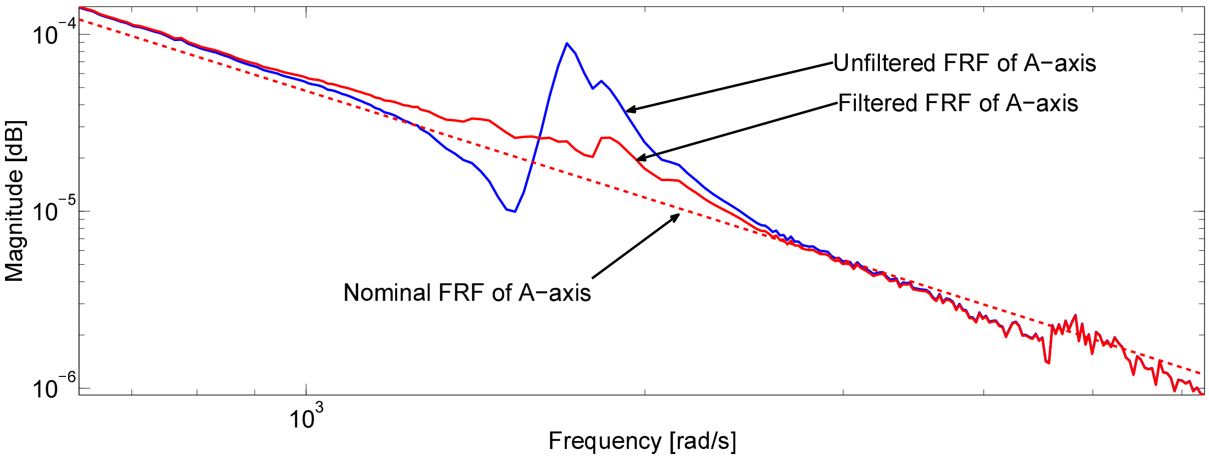


Figure 4.7: FRF of the a_f-axis including the nominal FRF and experimental FRFs with and without the notch filter

As it can be seen, though the experimental frequency response function (FRF) follows the nominal model at lower and higher frequencies, in between 1000 [rad/s] and 2000 [rad/s] resonant modes from the flexibilities of the rotary table can be seen. With the increased magnitude at approximately 1700 [rad/s], when the cross over frequency of the open loop transfer function was selected to be too high, the peak is pushed close to the 0dB reducing the gain margin and introducing oscillatory behaviour and potential instabilities. As a result, notch filters were implemented to remove the frequencies in the control signal that correspond to these resonant frequencies. The notch filters take the following form:

$$\begin{aligned} N_a(s) &= \frac{s^2 + 2\omega_a\zeta_{a,1} + \omega_a^2}{s^2 + 2\omega_a\zeta_{a,2} + \omega_a^2} \\ N_b(s) &= \frac{s^2 + 2\omega_b\zeta_{b,1} + \omega_b^2}{s^2 + 2\omega_b\zeta_{b,2} + \omega_b^2} \\ N(s) &= N_a(s)N_b(s) \end{aligned} \tag{4.76}$$

where N_a and N_b are the notch filters to remove the peak and valley, respectively, caused by the resonant mode. In the notch filters N_a and N_b, ω_a, ω_b are the resonance frequencies, selected based on peaks of the FRF, and (ζ_1, ζ_2) are the damping ratios, which are tuning parameters

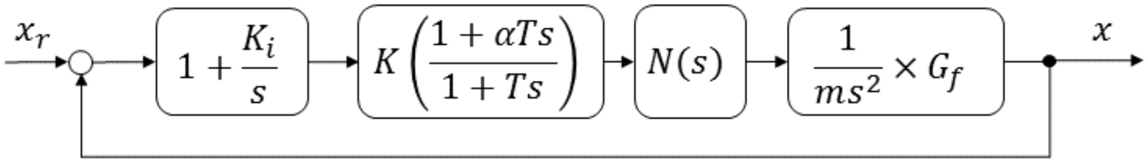

x_r
$1 + \frac{K_i}{s}$
$K\left(\frac{1+\alpha T s}{1+Ts}\right)$
$N(s)$
$\frac{1}{ms^2} \times G_f$
x

Figure 4.8: Block diagram of rotary table controller

Table 4.4: Rotary table parameters

Axis	x_f	y_f	z_f	a_f	b_f	c_f
J_q	2.3[kg]			20878[kg/mm^2]		40508 [kg/mm^2]
ω_m [rad/s]	1570.8					
K_f	5.95E5	6.60E5	8.14E5	3.32E3	2.24E3	6.53E3
α	1.31E2	1.31E2	1.31E2	1.31E2	1.31E2	3.75E2
T	5.57E-5	5.57E-5	5.57E-5	5.57E-5	5.57E-5	2.73E-5
K_i	157.1					
ω_a [rad/s]	-	-	1992	1705	1429	2582
$\zeta_{a,1}$	-	-	0.02	0.03	0.03	0.08
$\zeta_{a,2}$	-	-	0.3	0.3	0.2	0.02
ω_b [rad/s]	-	-	1740	1533	1300	2421
$\zeta_{b,1}$	-	-	0.3	0.3	0.2	0.05
$\zeta_{b,2}$	-	-	0.02	0.04	0.03	0.2

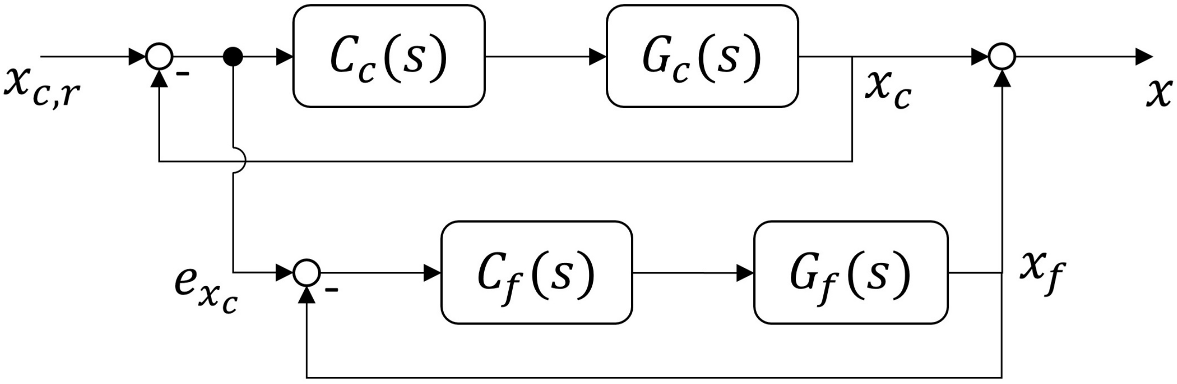


Figure 4.9: Dual stage feed drive controller

to achieve a desirable open loop transfer function. The full block diagram for the closed loop controller of the rotary table with the lead-lag compensator and notch filter is given in Fig. 4.8 and the parameters of the lead lag controllers and notch filters are shown in Tab. 4.4.

4.4 Dual-stage Feed Drive Control

Based on the above control designs, the achievable bandwidth of the micromill is 30Hz while the rotary table is 250Hz. Since the translational axes of the micromill and rotary table are parallel, the rotary table can be used to compensate for tracking errors coming from the lower bandwidth of the closed loop controller of the micromill. As a result, in addition to its own trajectory command, an additional input to the rotary table is the tracking error of the micromill as shown in Fig. 4.9.

In this configuration, the summation of the two parallel drives has the bandwidth of the rotary table, resulting in the long stroke of the micromill with the higher precision of the rotary table. The summation of the two drives gives the following transfer functions:

$$x = x_c + x_f \tag{4.77}$$

Based on the block diagram shown in Figure 4.9, Eq. (4.77) can be expressed as a function of the reference input as follows:

$$x = x_r \underbrace{\frac{C_c G_c}{(1 + C_c G_c)}}_{G_{c,cl}} + x_r \underbrace{\frac{1}{(1 + C_c G_c)}}_{S_{c,cl}} \underbrace{\frac{C_f G_f}{(1 + C_f G_f)}}_{G_{f,cl}} \tag{4.78}$$

where x_r is the reference command to the micromill feeddrive. As a result, as long as $G_{c,cl}$, $S_{c,cl}$, and $G_{f,cl}$ are stable then x will be stable. Since the micromill, $G_{c,cl}$ is stable, its tracking errors, which are used as input to the rotary table, $G_{f,cl}$, will be bounded and stable. Furthermore, the sensitivity transfer function of the coarse closed loop transfer function $S_{c,cl}$ will also be stable since $G_{c,cl}$ is designed to be stable. The two actuators are independently controlled and stable, and as a result the dual stage controller is also stable. The transfer function of the coupled system can also be expressed as follows:

$$\frac{x}{x_r} = \frac{C_c G_c + C_f G_f + C_c G_c C_f G_f}{(1 + C_f G_f)(1 + C_c G_c)} \tag{4.79}$$

At frequencies below the closed loop bandwidth of the micromill, ω_c, and rotary table, ω_f, $|G_c C_c| \gg 1$ and $|G_f C_f| \gg 1$. As a result the dual stage system has a closed loop gain of unity. However at frequencies greater than ω_c but below ω_f, $|G_c C_c| \leq 1$ and $|G_f C_f| \gg 1$ and as a result the closed loop dynamics of the summed response is as follows:

$$x \approx \frac{G_f C_f}{1 + G_f C_f} x_r \tag{4.80}$$

As a result, at frequencies above the design bandwidth of the closed loop controller for the micromill, the closed loop dynamics of the summed response adopts the closed loop dynamics of the rotary table. As a result, the summed system is able to have the stroke of the micromill but the precision of the rotary table. This can be seen in Fig. 4.10.

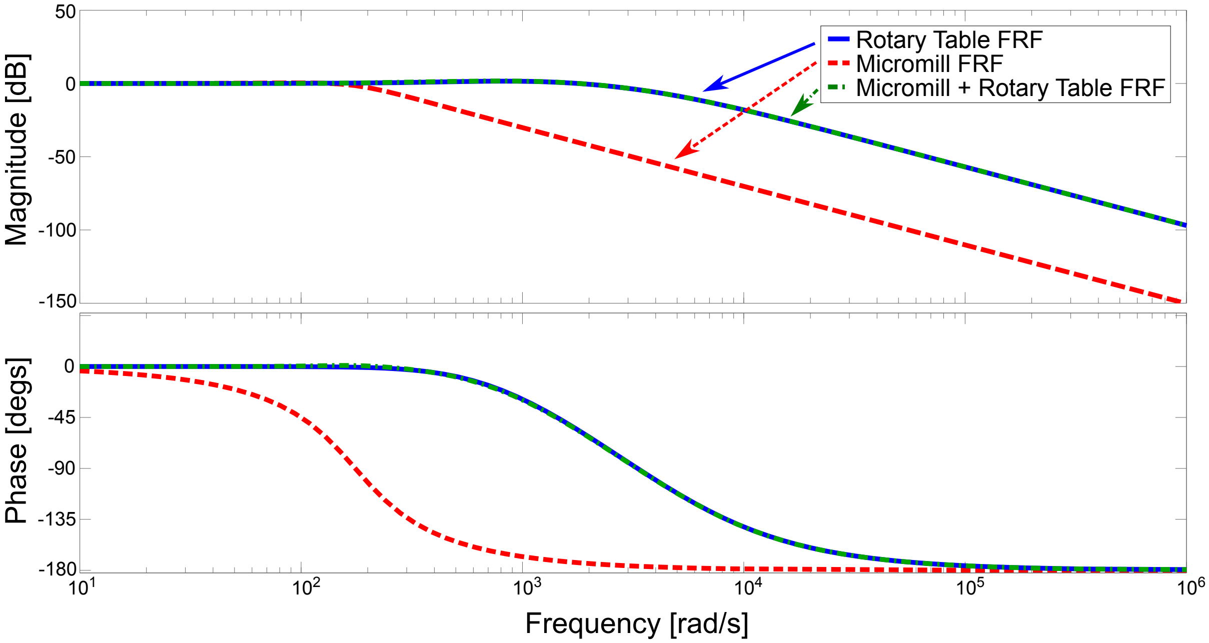


Figure 4.10: Simulated frequency response functions of micromill, rotary table, and micromill with tracking error compensated with rotary table.

When the rotary table is used to compensate for the tracking errors of the micromill, the transfer function of the axes adopts the bandwidth of the higher bandwidth controller. As a result, the dual stage feed drive system achieves the stroke of the micromill but with the high precision - high bandwidth control of the rotary table.

Chapter 5

Geometric Error Modeling for a 3-axis Micromill and Compensation with a 6 Degree of Freedom Rotary Table

5.1 Overview

Often in the discussion of the effectiveness of a manufacturing process, the major focus is on the precision of machine tool performing the manufacturing process. However, what is often less discussed is the accuracy of the manufacturing process. In most literature, it is assumed that accuracy follows precision. However this is typically not the case and additional methods must be implemented in order to increase the accuracy of the machine tool to a sufficient level, even at sufficient levels of precision. The source of the problem lies in the fact that most feedback servomechanisms are only capable of detecting errors in the direction of actuation. As a result, errors that are in directions orthogonal to the direction of travel can go undetected. These errors can originate from assembly errors, deflection caused by loading, or deviations caused by thermal expansion. Though these errors only create minimal problems for single axis of actuation, when multiple servomechanisms are combined and non-Cartesian movements are included, the geometric errors could result in large deviations from the desired end effector position and orientation when the whole kinematic model is considered.

In this chapter, the fine movements of the rotary table are used to compensate for tooltip errors caused by the assembly errors. Due to the invariant nature of errors caused by loading or thermal expansion, the rotary table would not be suitable for this task as these errors would change with operating parameters or environmental conditions. The ideal kinematic model from Chapter 3, is modified in Section 5.2 to include the effect of the geometric errors of each of the axis. Since the geometric errors originate from assembly errors of the axes, it changes consistently over the full actuating range of the axis. In Section 5.3, the geometric error is measured, then fit to a polynomial function of position. Finally, with the geometric errors known, a compensation algorithm is proposed in Section 5.4, where the rotary table position commands are modified to compensate for tooltip position errors.

5.2 Kinematic Model with Errors

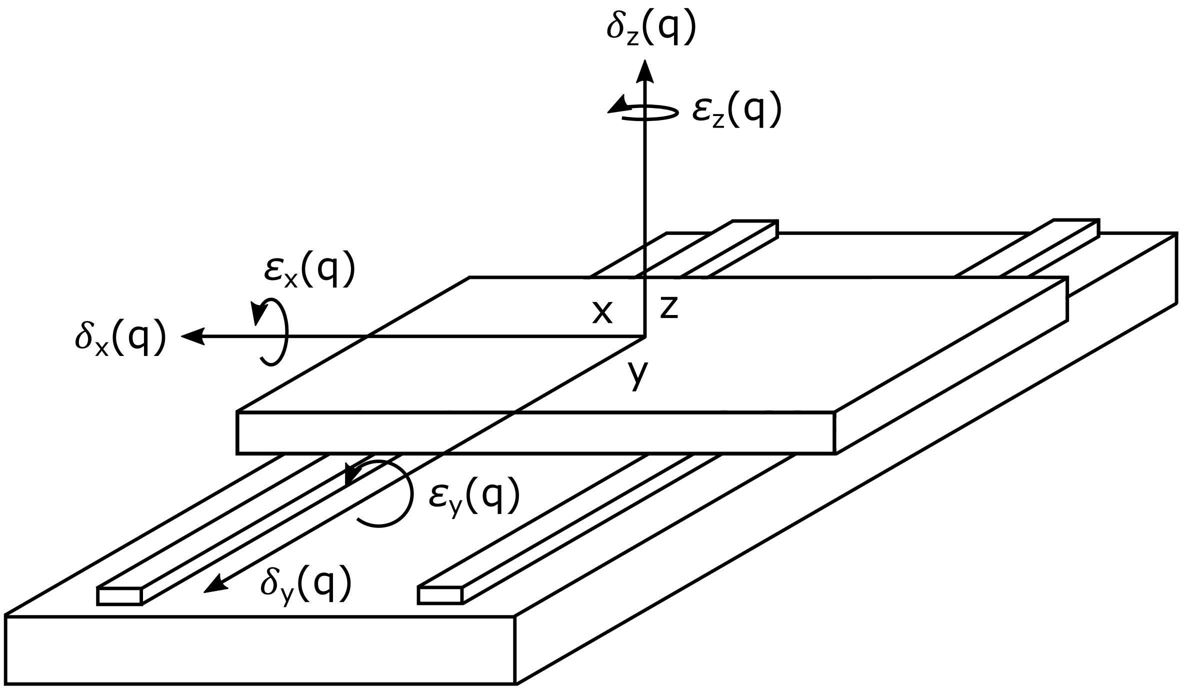


Figure 5.1: Geometric errors of a general axis q

A single axis has a total of 6 geometric errors, which include one positioning error, two straightness errors, a roll error, a pitch error, and a yaw error. These errors can be seen on a general axis, q, in Fig. 5.1.

The positioning error is typically caused by inaccuracies in the metrology system, while the other geometric errors are caused by assembly errors. Though the errors are small in magnitude, with multiple axes stacked and rotational errors being amplified with linear offsets, it is possible to have large resultant errors at the end effector. In order to demonstrate this, the kinematic model defined in Eq. (3.6) to (3.8), is modified to include the geometric errors [41]. The general error matrix is defined as follows:

$$E_q = \begin{bmatrix} 1 & -\varepsilon_z(q) & \varepsilon_y(q) & \delta_x(q) \\ \varepsilon_z(q) & 1 & -\varepsilon_x(q) & \delta_y(q) \\ -\varepsilon_y(q) & \varepsilon_x(q) & 1 & \delta_z(q) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$

where $\varepsilon_x(q)$, $\varepsilon_y(q)$, and $\varepsilon_z(q)$ are the rotations about the x, y, and z axis of the current coordinate frame, respectively, and $\delta_x(q)$, $\delta_y(q)$, and $\delta_z(q)$ are displacements in the x, y, and z axis of the current coordinate frame, respectively, as shown in Fig. 5.1. It should be noted that Eq.(5.1) uses the small angle approximation for the cosine and sine of the rotational errors, ε, since the rotational errors are typically sufficiently small. By multiplying Eq.(5.1) with the ideal HTM, it is possible to project the effect of errors in one axis onto the next axis, and subsequently, to the tooltip. For example, transforming from base coordinate frame to the y-axis of the micromill has an ideal transformation as follows:

$$^{B}\mathbf{T}_{y_c,i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y_c \\ 0 & 0 & 1 & L_{y,z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

However, when the error matrices are multiplied, the actual transformation becomes as follows:

$$\begin{aligned} {}^{B}\mathbf{T}_{y_c,a} &= {}^{B}\mathbf{T}_{y_c,i}E_{y_c} \\ &= \begin{bmatrix} 1 & -\varepsilon_z(y_c) & \varepsilon_y(y_c) & \delta_x(y_c)+y_c \\ \varepsilon_z(y_c) & 1 & -\varepsilon_x(y_c) & \delta_y(y_c) \\ -\varepsilon_y(y_c) & \varepsilon_x(y_c) & 1 & \delta_z(y_c)+L_{y,z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{5.3}$$

As a result, in order to model the effect of the geometric errors of each axis on the position of the tooltip, the ideal transformation matrices are multiplied with their respective error transformation matrices, making changes to the two kinematic chains in Eq.(3.1) and Eq.(3.2) as follows:

$$\begin{aligned} {}^{B}\mathbf{T}_{w,a} &= {}^{B}\mathbf{T}_{y_c,i}E_{y_c}{}^{y_c}\mathbf{T}_{x_c,i}E_{x_c}{}^{x_c}\mathbf{T}_{s,i}{}^{s}\mathbf{T}_{x_f,i} \\ &\quad\times{}^{s}\mathbf{T}_{x_f,i}E_{x_f}{}^{x_f}\mathbf{T}_{y_f,i}E_{y_f}{}^{y_f}\mathbf{T}_{z_f,i}E_{z_f} \\ &\quad\times{}^{z_f}\mathbf{T}_{a_f,i}E_{a_f}{}^{a_f}\mathbf{T}_{b_f,i}E_{b_f}{}^{b_f}\mathbf{T}_{c_f,i}E_{c_f} \\ &\quad\times{}^{c_f}\mathbf{T}_{r,i}{}^{r}\mathbf{T}_{w,i} \end{aligned} \tag{5.4}$$

$${}^{B}\mathbf{T}_{t,a} = {}^{B}\mathbf{T}_{z_c,i}E_{z_c}{}^{z_c}\mathbf{T}_{t,i} \tag{5.5}$$

which will give the actual HTM from the tool coordinate frame to the workpiece coordinate frame:

$$\mathbf{T}_a = \left({}^{B}\mathbf{T}_{w,a}\right)^{-1}{}^{B}\mathbf{T}_{t,a} \tag{5.6}$$

With the actual kinematic transformation defined in Eq.(5.6), the tooltip position with the effect of error is transformed to a position that is with respect to the workpiece with a method similar to Eq.(3.5):

$$\begin{bmatrix} P_{x,a} \\ P_{y,a} \\ P_{z,a} \\ 1 \end{bmatrix} = \mathbf{T}_a \begin{bmatrix} {}^{T}\mathbf{t}_p \\ 1 \end{bmatrix} \tag{5.7}$$

which gives the actual tooltip position, $\mathbf{P}_a\left(\mathbf{q}\right) = \left[P_{x,a}\left(\mathbf{q}\right), P_{y,a}\left(\mathbf{q}\right), P_{z,a}\left(\mathbf{q}\right)\right]^T$. The result of the matrix multiplication in Eq. (5.7), with second order and higher terms error grouped as $\mathcal{O}(\varepsilon^2, \delta^2, \varepsilon\delta)$, can be found in Appendix A. The forward kinematic model now also includes the effect of the geometric errors on the tooltip as a function of axes positions. In this work, the forward kinematic equations consists of 9-axis commands and 18 position dependent error terms. As a result, the equations can become very long, and subsequently, unmanageable and computationally expensive. In order to overcome this problem, the second order or higher terms involving geometric errors are approximated as follows:

$$\mathcal{O}(\varepsilon^2, \delta^2, \varepsilon\delta) \approx 0 \quad \text{for} \quad \varepsilon, \delta \to 0 \tag{5.8}$$

since the geometric errors are very small relative to the position commands and offsets. Given this approximation, the forward kinematic equations can be simplified and approximated by including only the zero and first order terms of geometric errors in the forward kinematic equations. With the kinematic equations modified to include the geometric errors, the position dependent error equations must be identified through experimentation.

5.3 Error Modeling

From Eq. (A.1), Eq. (A.2), and Eq. (A.3), it can be seen that the geometric errors play a strong role in determining the resultant position of the tooltip position. As a result these errors must be measured and fit to a function of position in order to be incorporated into the kinematic model. In order to do this, the errors of the machine tool are first measured with a laser interferometer using various mirror configurations to get geometric errors in different directions. Since generated trajectories are on average jerk continuous [51], a quintic polynomial is selected as the function to be fit to the error data in order to maintain the same jerk continuity. For each geometric error, there should be $N_\epsilon \times M_\epsilon$ corresponding error measurements, ϵ, where N_ϵ is the total number of measurement points on the axis and M_ϵ is the number of repeated measurements at each position. In this work, the errors are measured 6 times, every 1[mm] and the measurement ϵ can take the form of a displacement error, δ, or a rotational error ε. The objective is to find a quintic polynomial that best fits this data. First the M_ϵ repeated measurements at N_ϵ locations are averaged:

$$\epsilon^*_{n_\epsilon} = \frac{\epsilon_{1,n_\epsilon} + \epsilon_{2,n_\epsilon} + \cdots + \epsilon_{M_\epsilon,n_\epsilon}}{M_\epsilon} \tag{5.9}$$

where $n_\epsilon = 1, 2, \ldots, N_\epsilon$. With the averaged errors, the objective is to fit a quintic polynomial to the data points with as little error as possible. The quintic polynomial fitting problem can be phrased as follows:

$$\underbrace{\begin{bmatrix} \epsilon^*_1 \\ \epsilon^*_2 \\ \vdots \\ \epsilon^*_{N_\epsilon-1} \\ \epsilon^*_{N_\epsilon} \end{bmatrix}}_{\mathbf{Y}_\epsilon} = \underbrace{\begin{bmatrix} q_1^5 & q_1^4 & q_1^3 & q_1^2 & q_1 & 1 \\ q_2^5 & q_2^4 & q_2^3 & q_2^2 & q_2 & 1 \\ \vdots & & \ddots & & & \vdots \\ q^5_{N_\epsilon-1} & q^4_{N_\epsilon-1} & q^3_{N_\epsilon-1} & q^2_{N_\epsilon-1} & q_{N_\epsilon-1} & 1 \\ q^5_{N_\epsilon} & q^4_{N_\epsilon} & q^3_{N_\epsilon} & q^2_{N_\epsilon} & q_{N_\epsilon} & 1 \end{bmatrix}}_{\Phi_\epsilon} \underbrace{\begin{bmatrix} \varphi_A \\ \varphi_B \\ \varphi_C \\ \varphi_D \\ \varphi_E \\ \varphi_F \end{bmatrix}}_{\theta_\epsilon} \tag{5.10}$$

where $\mathbf{Y}_\epsilon$ is a vector of averaged measured geometric errors, Φ_ϵ is a matrix of the axis positions where the laser measurements took place, and θ_ϵ is a vector of coefficients for the fitting polynomial . The objective is to find the coefficients in θ_ϵ that minimizes the mean squared error between the predicted geometric error and actual measured geometric error or more formally:

$$\min \frac{1}{2} (\mathbf{Y}_\epsilon - \Phi_\epsilon \theta_\epsilon)^T (\mathbf{Y}_\epsilon - \Phi_\epsilon \theta_\epsilon) \tag{5.11}$$

This is done in a least squares fashion in the following way:

$$\theta_\epsilon = (\Phi_\epsilon^T \Phi_\epsilon)^{-1} \Phi_\epsilon^T \mathbf{Y}_\epsilon \tag{5.12}$$

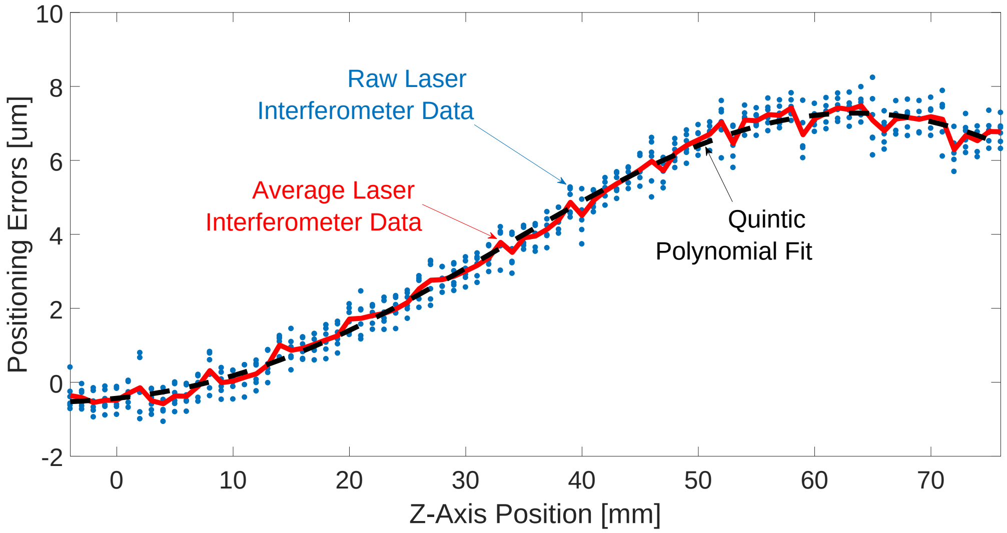


Figure 5.2: z_c-axis positioning errors and the resultant quintic polynomial fit

and as a result the equation of error becomes

$$\hat{\epsilon}\left(q\right) = \varphi_A q^5 + \varphi_B q^4 + \varphi_C q^3 + \varphi_D q^2 + \varphi_E q^1 + \varphi_F \tag{5.13}$$

where $\hat{\epsilon}$ indicates the approximated geometric error. This process is repeated for all 6 errors on each of the major axes giving a total of up to 18 quintic polynomial functions. An example of this process can be seen in Fig. 5.2 where the positioning errors of the z_c-axis are fit to a quintic polynomial.

Once all the geometric errors are measured and fitted, the errors can be included into Eq. (A.1), Eq. (A.2), and Eq. (A.3). As a result, the actual tooltip position is known and its errors can be compensated.

5.4 Geometric error compensation

Once the geometric errors of each of the three axes are known, the actual tooltip position is known. Subsequently it is possible to find the error of the tooltip position from the desired tooltip position by subtracting the actual tooltip position from the desired tooltip position as follows:

$$\Delta\mathbf{P}\left(\mathbf{q}\right) = \left|\mathbf{P}_i - \mathbf{P}_a(\mathbf{q})\right| \tag{5.14}$$

The ideal tooltip position $\mathbf{P}_i$ can be provided from a computer aided manufacturing (CAM) toolpath. Alternatively, if only the axes positions are known $\mathbf{P}_i$ can also be the result of the ideal forward kinematic model defined in Eq. (3.6) to Eq. (3.8), prior to the reference command modifications performed for geometric error compensation. The objective is to find rotary table reference commands, $\mathbf{q}_f = [x_f, y_f, z_f, a_f, b_f, c_f]^T$, that will reduce the geometric error to an acceptable minimum. In order to do this, two challenges exist. First, similar to the trajectory generation of 9-axis machine tool, there are more degrees of freedom available than necessary. Since only the tooltip position is defined with 3 degrees of freedom and the rotary table has 6 degrees of freedom, there are 3 redundant degrees of freedom. As a result, redundancies must be resolved with the proposed geometric error compensation technique. Second, due to the non-Cartesian kinematics of the overall 9-axis configuration, the compensation of geometric errors is non-trivial and numerical techniques are necessary. To overcome these challenges, an iterative gradient descent algorithm which exploits the Jacobian matrix of the rotary table is proposed.

5.4.1 Gradient Descent Optimization Algorithm Background

The gradient descent optimization algorithm is a first order iterative optimization algorithm that is used for finding the minimum of a function. The algorithm iterates by selecting function variables that are proportional to the negative of the gradient of the function at the current point. This process is repeated until the gradient is sufficiently small, which corresponds to a local minimum of the function. If the variables to be modified are defined as η and the differentiable multivariable cost function is defined as $\Psi(\eta)$ then the cost decreases fastest if η goes in the direction of the negative gradient of the cost function as follows:

$$\eta_{j+1} = \eta_j - \gamma \nabla \Psi(\eta_j) \tag{5.15}$$

where j is the current number of iteration for the gradient descent algorithm. For sufficiently small γ the cost will decrease at each iteration. As a result, starting with a guess of η_0 the cost will decrease as follows:

$$\Psi(\eta_0) \geq \Psi(\eta_1) \geq \Psi(\eta_2) \geq \dots \tag{5.16}$$

5.4.2 Geometric Error Compensation using Gradient Descent Optimization Algorithm

With respect to geometric error compensation, the gradient descent algorithm is used to modify the rotary table commands, $\mathbf{q}_f$, so that the tooltip position errors $\Delta\mathbf{P}\left(\mathbf{q}_f\right)$ are decreased to an acceptable minimum. As a result the variables to be modified are $\mathbf{q}_f$ and the cost function is defined to penalize tooltip errors as follows:

$$\Psi\left(\mathbf{q}_f\right) = \frac{1}{2}\Delta\mathbf{P}\left(\mathbf{q}_f\right)^T \Delta\mathbf{P}\left(\mathbf{q}_f\right) \tag{5.17}$$

When applied to the problem of geometric error compensation, the gradient descent algorithm can be phrased as follows:

$$\mathbf{q}_{f,j+1} = \mathbf{q}_{f,j} - \nabla\Psi\left(\mathbf{q}_{f,j}\right) \tag{5.18}$$

where γ is set to 1, j is the number of iterations of the gradient descent algorithm and the gradient of the cost function Ψ is:

$$\nabla\Psi\left(\mathbf{q}_f\right) = \frac{d\mathbf{q}_f}{d\mathbf{P}_a}\Delta\mathbf{P}\left(\mathbf{q}_f\right) \tag{5.19}$$

Since, $\Delta\mathbf{P}\left(\mathbf{q}_f\right)$ is known from Eq. (5.14), the differential term $\frac{d\mathbf{q}_f}{d\mathbf{P}_a}$ must be defined in order to obtain the gradient. Since there are more degrees of freedom than necessary, an infinite range of joint configurations are possible, or more formally, $\frac{d\mathbf{q}_f}{d\mathbf{P}_a}$ is non unique.

Similar to the redundancy resolution technique presented in Chapter 3, the Moore-Penrose inverse is used to resolve the redundancies. Since the geometric error compensation only uses the rotary table commands, the Jacobian used for compensation will only consider the effect of the rotary table commands on the tooltip position and tool orientation. Furthermore, the Jacobian will also incorporate the geometric errors in the forward kinematics equation. As a result, the compensating Jacobian is defined as follows:

$$J^c = \frac{d\mathbf{P}_a}{d\mathbf{q}_f} = \begin{bmatrix} \frac{dP_{x,a}}{dx_f} & \cdots & \frac{dP_{x,a}}{dc_f} \\ \frac{dP_{y,a}}{dx_f} & \cdots & \frac{dP_{y,a}}{dc_f} \\ \frac{dP_{z,a}}{dx_f} & \cdots & \frac{dP_{z,a}}{dc_f} \end{bmatrix}_{3\times 6} \tag{5.20}$$

where the superscript c differentiates the Jacobian defined in Eq. (5.20) from the Jacobian defined in Eq. (3.13). With the compensating Jacobian, J^c, a Moore-Penrose inverse can be found as a possible solution to $\frac{d\mathbf{q}_f}{d\mathbf{P}_a}$

$$J^{c\dagger} = \frac{J^{cT}}{J^c J^{cT}} \tag{5.21}$$

As a result, the gradient of the objective function can be redefined as follows:

$$\nabla\Psi\left(\mathbf{q}_{f,j}\right) = J^{c\dagger}\Delta\mathbf{P}\left(\mathbf{q}_{f,j}\right) \tag{5.22}$$

However, in addition to minimizing the error between the desired tooltip position and the actual tooltip position, the algorithm must ensure that the generated compensation commands do not violate the stroke limits of the rotary table. Similar to the trajectory generation algorithm presented in Chapter 3, an additional constraint can be projected onto the nullspace of $J^{c\dagger}$. The cost function which constraints the magnitude of the stroke limits is considered as follows:

$$H^c\left(\mathbf{q}_f\right) = \frac{x_f^2}{\left(x_{f,max}^2 - x_f^2\right)^2} + \cdots + \frac{c_f^2}{\left(c_{f,max}^2 - c_f^2\right)^2} \tag{5.23}$$

By projecting the gradient of this cost function:

$$\nabla H^c\left(\mathbf{q}_f\right) = \left[\frac{2x_{f,max}^2 x_f}{\left(x_{f,max}^2 - x_f^2\right)^2}, \ldots, \frac{2c_{f,max}^2 c_f}{\left(c_{f,max}^2 - c_f^2\right)^2}\right] \tag{5.24}$$

onto the nullspace of $J^{c\dagger}$, it is ensured that the solutions for $\nabla\Psi\left(\mathbf{q}_f\right)$ will not violate the stroke limits of the rotary table. It should be noted that the constraint equation, Eq. (5.23), and its gradient, Eq. (5.24), only consider the limits of the rotary table. Since only the reference commands of the rotary table are being modified, it is suffice to only consider the rotary table position commands. The gradient of the cost function is extended to consider this additional constraint in the following way:

$$\nabla\Psi\left(\mathbf{q}_f\right) = J^{c\dagger}\Delta\mathbf{P}\left(\mathbf{q}_f\right) + \left(I - J^{c\dagger}J^c\right)\beta^c\nabla H^c\left(\mathbf{q}_f\right) \tag{5.25}$$

where β^c determines the strength of the stroke limit constraint defined in H^c. The gradient descent formulation can be redefined as follows:

$$\mathbf{q}_{f,j+1} = \mathbf{q}_{f,j} - J^{c\dagger}\Delta\mathbf{P}\left(\mathbf{q}_{f,j}\right) - \left(I - J^{c\dagger}J^c\right)\beta^c\nabla H^c\left(\mathbf{q}_{f,j}\right) \tag{5.26}$$

Since the the cost, $\Psi\left(\mathbf{q}_{f,j}\right)$, decreases the fastest when the rotary table commands, $\mathbf{q}_{f,j+1}$, are recalculated in the direction of $-\nabla\Psi\left(\mathbf{q}_{f,j}\right)$ then every iteration will minimize the cost. As a result, Eq.(5.27) is iterated until the following condition is met:

$$\Delta\mathbf{P}\left(\mathbf{q}_f^*\right) < \nu \tag{5.27}$$

Algorithm 2: Full compensation algorithm

```
function Compensation ($\mathbf{q}$);
Input  : $\mathbf{Q} = [\mathbf{q}[0], \mathbf{q}[1], \ldots, \mathbf{q}[K-1], \mathbf{q}[K]]$
Output: $\mathbf{Q}^* = [\mathbf{q}_f^*[0], \mathbf{q}_f^*[1], \ldots, \mathbf{q}_f^*[K-1], \mathbf{q}_f^*[K]]$
$k = 0$;
$j = 0$;
for $k \leftarrow 0$ to $K-1$ do
    $\Delta\mathbf{P} = |\mathbf{P}_i(\mathbf{q}[k]_0) - \mathbf{P}_a(\mathbf{q}[k]_0)|$ ;
    while $\Delta\mathbf{P}\left(\mathbf{q}[k]_j\right) > e$ do
        $\mathbf{q}[k]_{j+1} = \mathbf{q}[k]_j - J^{c\dagger}\left(\mathbf{q}[k]_j\right)\Delta\mathbf{P}\left(\mathbf{q}[k]_j\right) - \left(I - J^{c\dagger}J^c\right)\beta^c\nabla H^c\left(\mathbf{q}_j\right)$ ;
        $\Delta\mathbf{P} = |\mathbf{P}_i(\mathbf{q}[k]_0) - \mathbf{P}_a(\mathbf{q}[k]_{j+1})|$ ;
        $j = j + 1$
    end
    $\mathbf{q}^*[k] = \mathbf{q}[k]_j$
    $j = 0$
    $k = k + 1$
end
```

where $\mathbf{q}_f^*$ is the rotary table commands that reduce the geometric errors to an acceptable limit defined by ν. Compensation commands are calculated for every point, $\mathbf{q}[k]$, which is known from the trajectory algorithm. The full algorithm can be summarized in Algorithm 2.

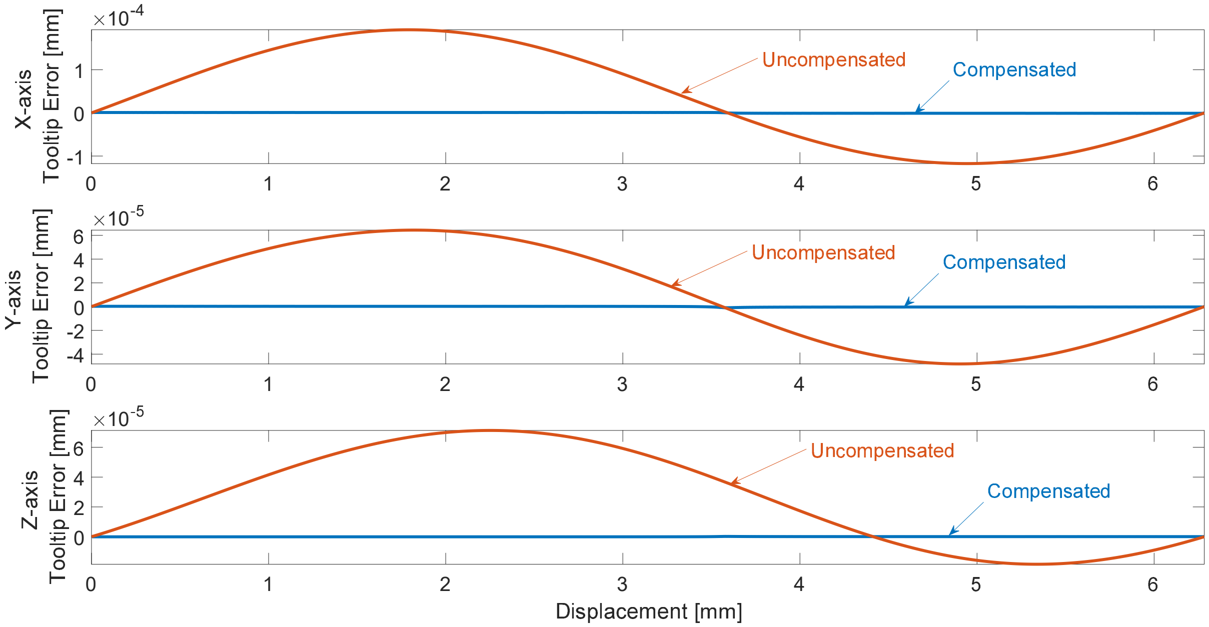


Figure 5.3: Tooltip errors with and without compensation for a circle on the x-y plane of radius 1

Simulation results using this method can be seen in Fig. 5.3, where the algorithm eliminates the tooltip errors caused by the geometric errors of the machine tool when drawing a circle of radius 1 [mm] on the x-y plane.

Chapter 6

Simulation and Experimental Results

6.1 Overview

Experiments were performed to validate the trajectory generation algorithm, dual stage feed drive tracking control law, and geometric error compensation method presented in this thesis. All experiments were performed on the machine presented in the Introduction.

6.2 Trajectory Generation Experimental Results

In order to validate the trajectory generation algorithm, two experiments were performed to ensure that the generated trajectories were able to resolve redundancies, avoid singularities, respect the prescribed limits, and follow the desired end effector trajectory. First, the trajectory generation algorithm is used to generate axes commands for a spiral toolpath as shown in Fig. 6.1

The position commands for the spiral toolpath and the time derivatives toolpath are shown in Fig. 6.2, Fig. 6.3, Fig. 6.4, and Fig. 6.5. As it can be seen, the position commands respect the stroke limits of all the drives, demonstrating the ability of the redundancy resolution component of the trajectory generation algorithm to select axes configurations that respect the stroke limits of the machine tool. In addition to this it can be seen that the generated axes commands respect the prescribed velocity, acceleration, and jerk limits of all the axes showing successful integration of the redundancy resolution algorithm with the feed optimization algorithm. It should also be noted that at certain portions of the trajectory, the velocity of the y_c is saturated

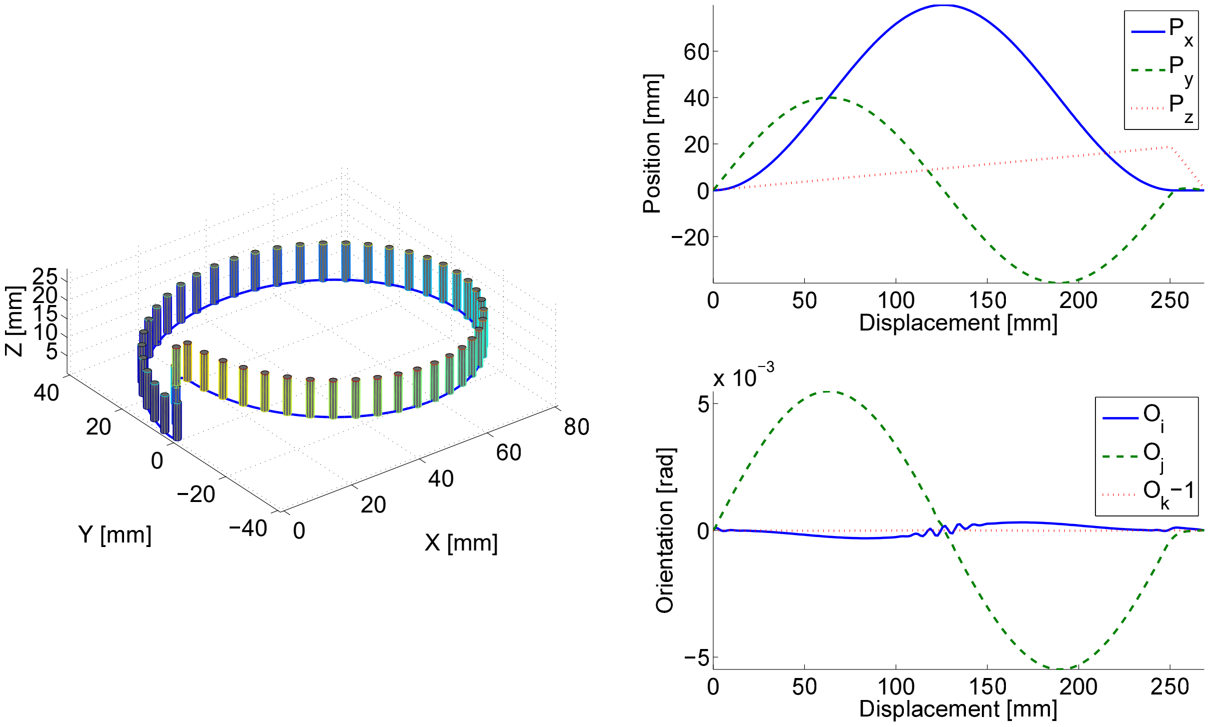


Figure 6.1: Spiral toolpath

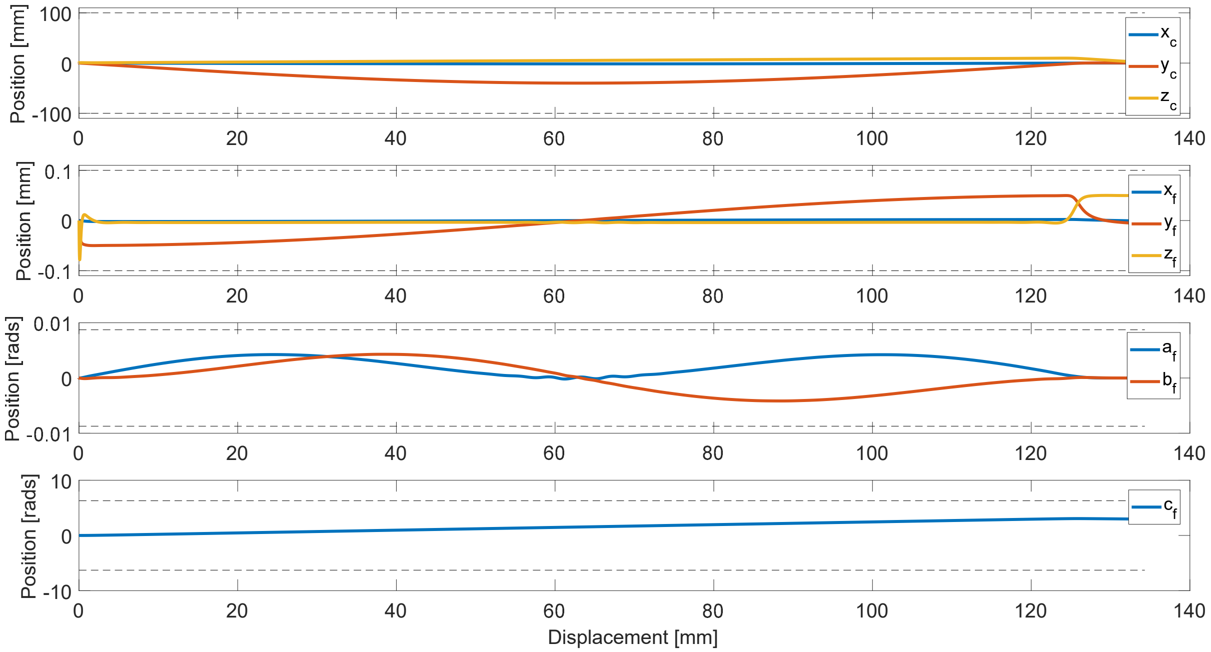


Figure 6.2: Reference commands for spiral trajectory

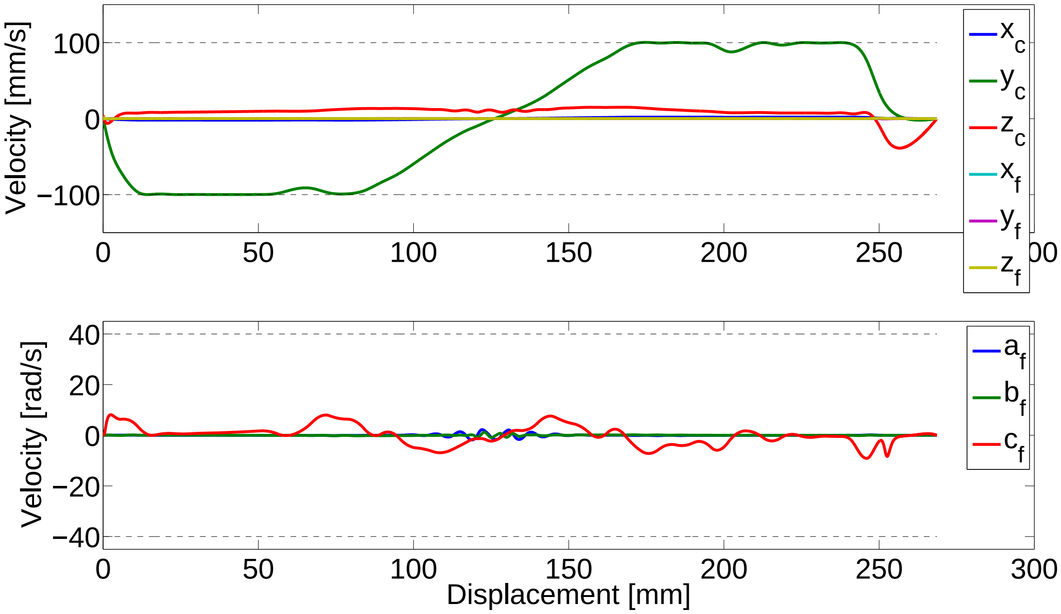


Figure 6.3: Velocity of reference commands for spiral trajectory

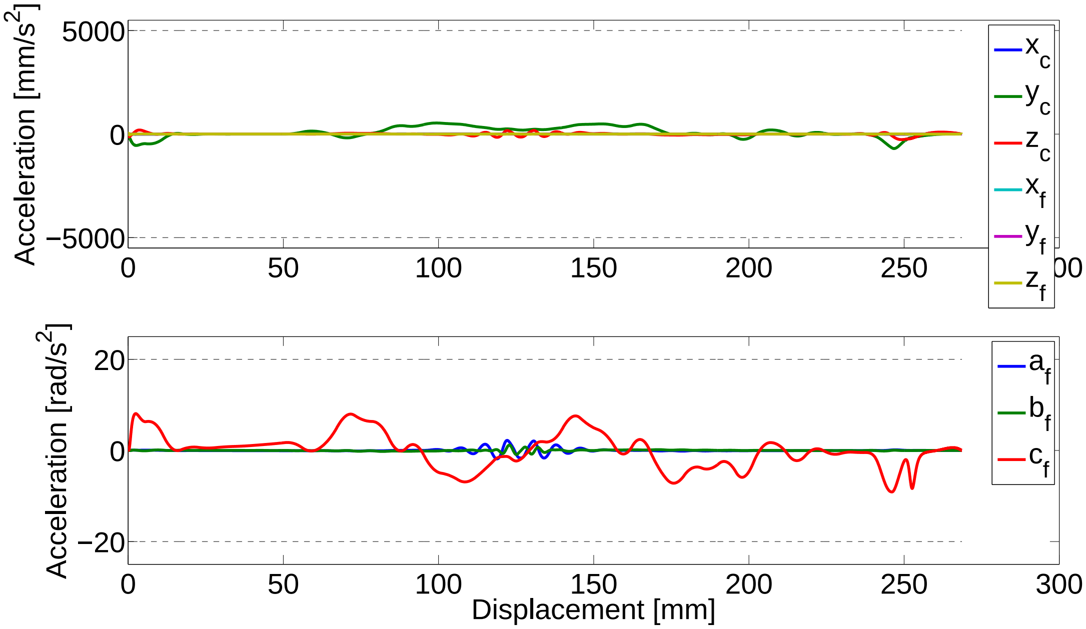


Figure 6.4: Acceleration of reference commands for spiral trajectory

at its limits, and for portions of the trajectory where the velocity is not saturated, the jerk of the other axes are close to being saturated. This demonstrates the ability of the feed optimization algorithm to use all the kinematic limits of the feed drive to traverse the toolpath.

In addition to respecting the limits of the axes, the generated axes commands should result in a toolpath that follows the desired toolpath. In the case of the trajectory generation algorithm presented, the numerical nature of the trajectory generation algorithm and singularity avoidance components can cause the toolpath from the axes commands to deviate from the desired toolpath. As a result, the desired tooltip position and tool orientation are compared with the tooltip position and tool orientation from applying forward kinematics on the generated axes position commands. The tool deviations for the spiral toolpath were shown earlier in Fig. 3.7 showing negligible deviation, and as a result, successful implementation of the a higher order

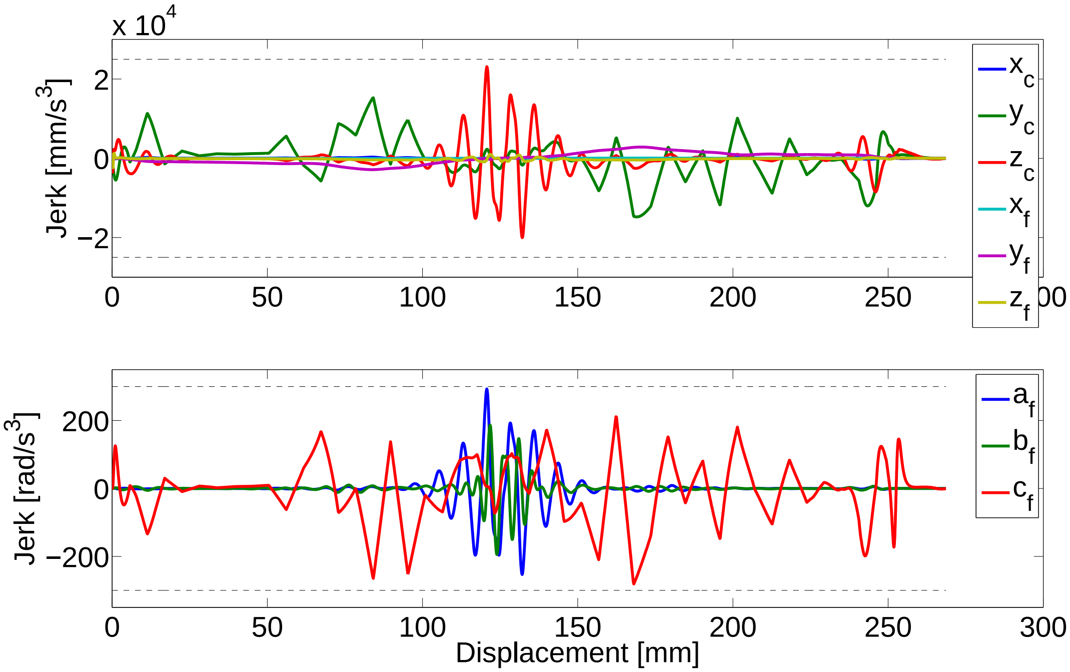


Figure 6.5: Jerk of reference commands for spiral trajectory

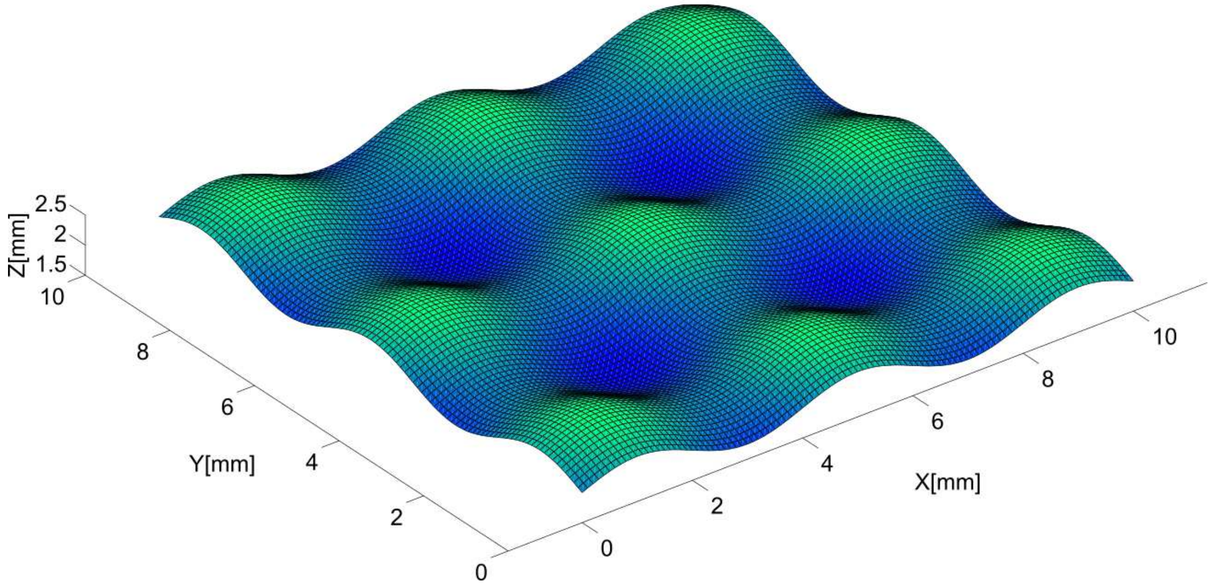


Figure 6.6: Sinusoidal freeform surface

numerical integration algorithm and closed loop corrective action.

Next the trajectory generation algorithm is used to generate reference commands to machine the freeform surface shown in Fig. 6.6.

The freeform surface is a sinusoidal surface with a peak to peak amplitude of 500 microns and a working surface area of 10 [mm] x 10 [mm]. This surface was cut with a 1/64" (397[μm]) Mitsubishi Carbide 2-fluted ball endmill. The finishing pass was done with a feedrate of 10 [mm/s] with a spindle speed of 170000 [rev/min]. To avoid the ploughing affect from the zero cutting velocity from the tip of the ball end mill, the tool is commanded to have a constant tilt of 0.15 degrees using the magnetically levitate table's six degrees of freedom. Since this was a finishing operation, the limitation of the feedrate will also come from the process itself, which was set at 10 [mm/s], in addition to the axes limits. The freeform surface was cut with a zig-

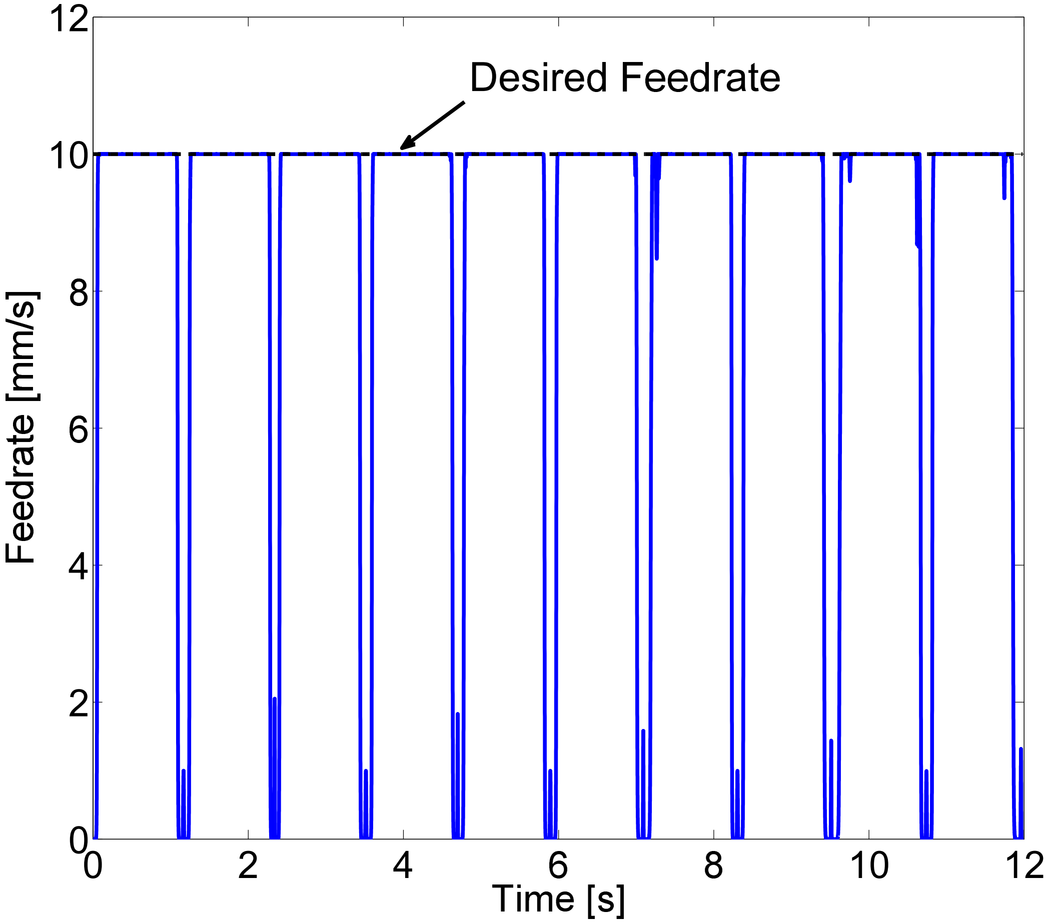


Figure 6.7: Feedrate profile for freeform surface with limit set to 10 [mm/s]

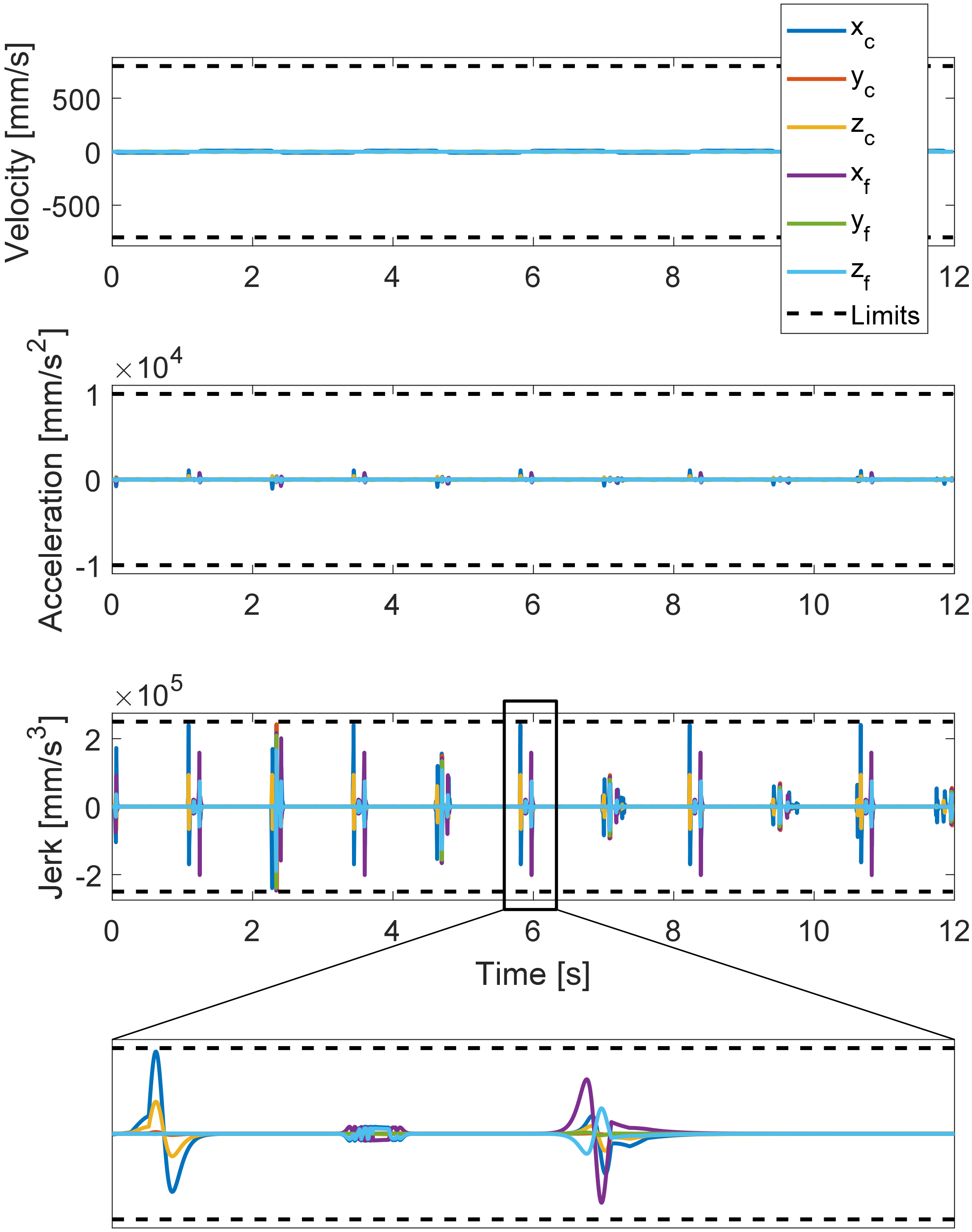


Figure 6.8: Velocity, acceleration, and jerk profile of the translational axes for the freeform surface

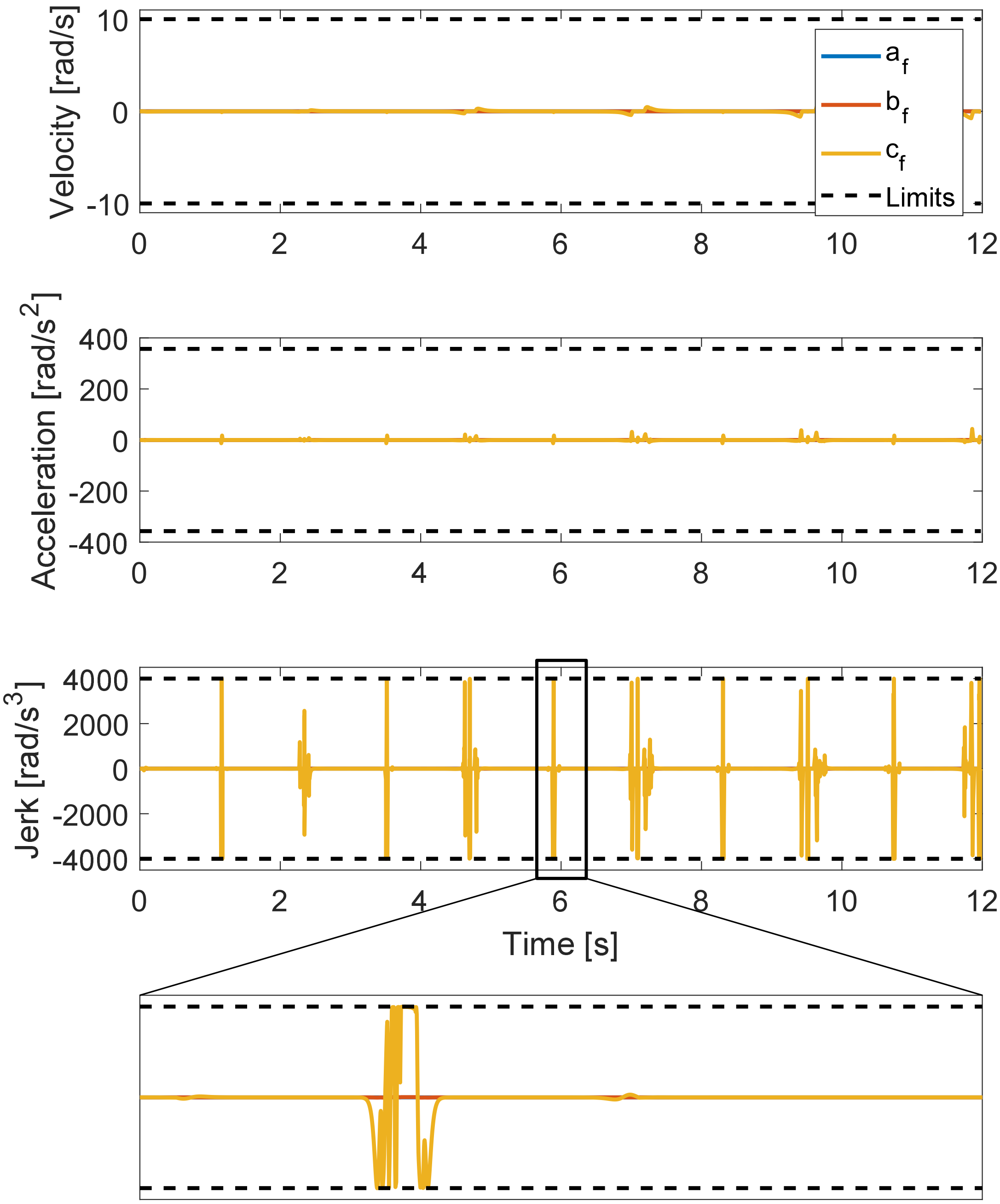


Figure 6.9: Velocity, acceleration, and jerk profile of the rotational axes for the freeform surface

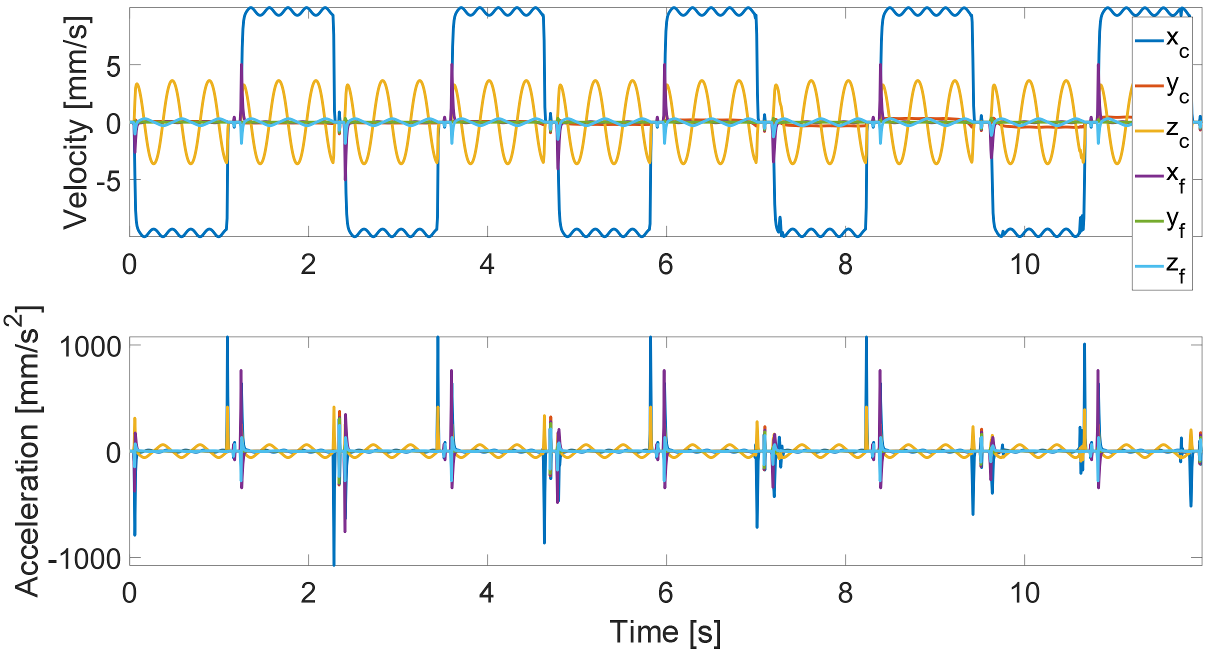


Figure 6.10: Velocity, acceleration, and jerk profile of the translational axes for the freeform surface

zag toolpath with a depth of cut of 100 [μm] and a step over of 10 [μm]. The planned feedrate is shown in Fig. 6.7 and as it can be seen, the feedrate is capped at 10 [mm/s]. However, at portions where the tool path must slow down and speed up, the limitation comes from the axes. This can be seen in the velocity, acceleration, and jerk of the reference commands in Fig. 6.8 and Fig. 6.9 for the translational and rotational axes respectively

Furthermore, it should be noted that when zoomed in, the velocity and acceleration profiles are smooth and continuous as shown in Fig. 6.10 and Fig. 6.11 for the translational and rotational axes respectively

The manufactured surface can be seen in Fig. 6.12. As it can be seen, the produced part matches the desired surface, showing the kinematic model is correct. Due to the finite step

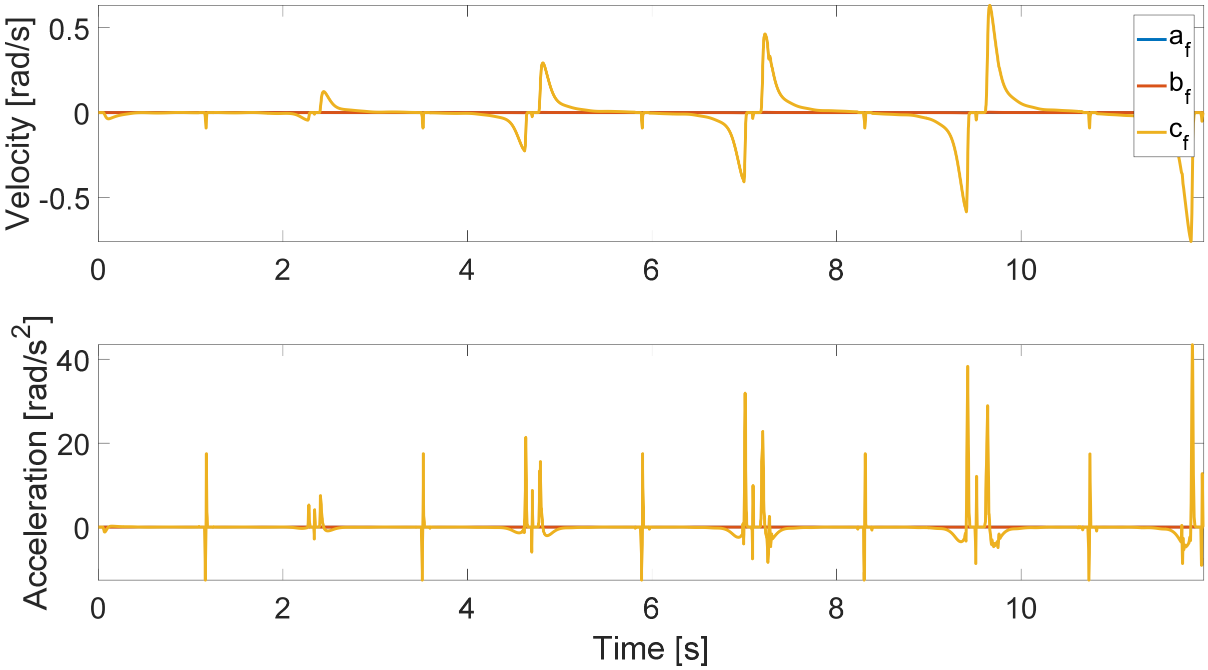


Figure 6.11: Velocity, acceleration, and jerk profile of the rotational axes for the freeform surface

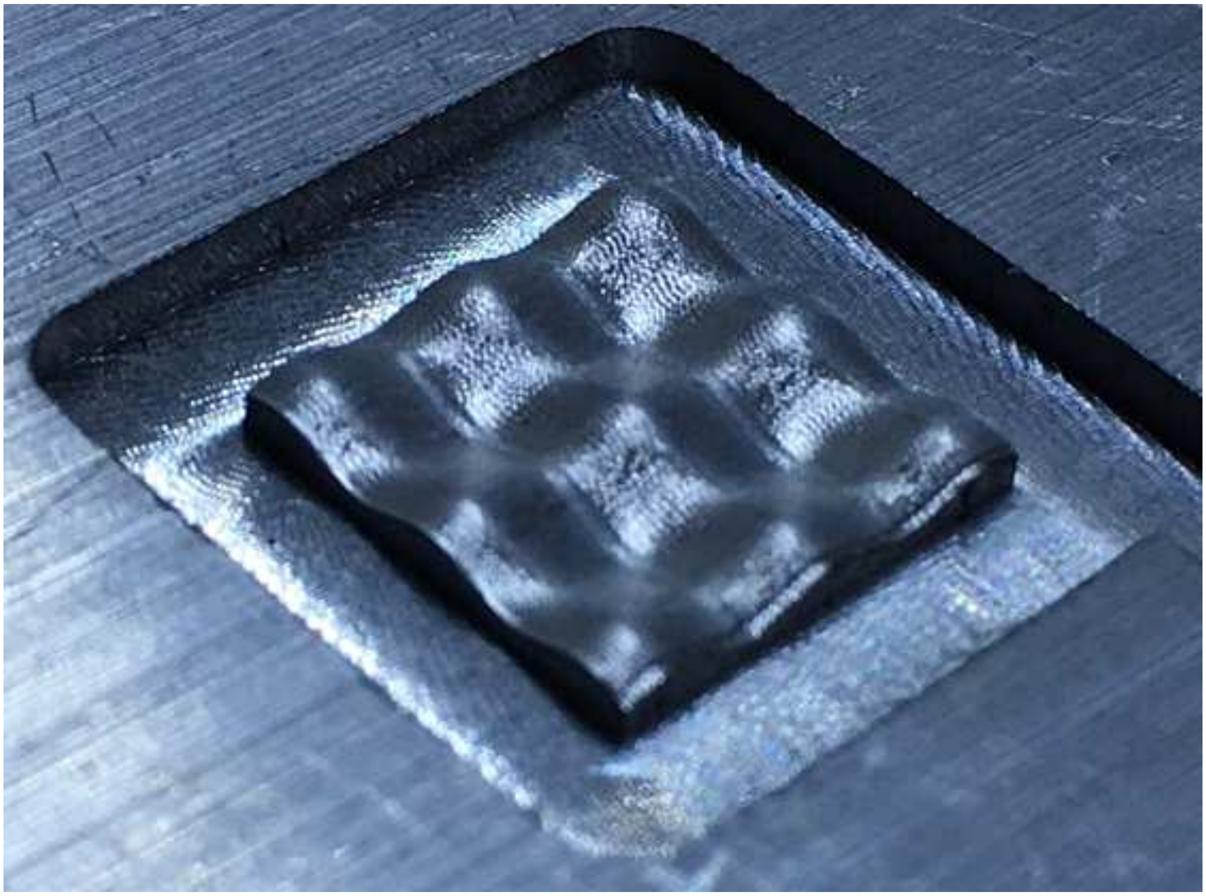


Figure 6.12: Machined freeform surface

Table 6.1: Circular contouring results

Error	SMC	DSFD
Mean | x-axis Tracking | [mm]	2.14E-4	1.98E-4
Mean | y-axis Tracking | [mm]	1.06E-4	4.94E-5
Mean Contour [mm]	1.58E-4	1.21E-4
Max | x-axis Tracking | [mm]	2.67E-3	1.28E-3
Max | y-axis Tracking | [mm]	1.72E-3	3.77E-4
Max Contour [mm]	2.67E-3	8.58E-4

Table 6.2: Square contouring results

Error	SMC	DSFD
Mean | x-axis Tracking | [mm]	2.58E-4	1.69E-4
Mean | y-axis Tracking | [mm]	8.81E-5	7.18E-5
Mean Contour [mm]	7.92E-5	9.93E-5
Max | x-axis Tracking | [mm]	5.43E-3	2.26E-3
Max | y-axis Tracking | [mm]	1.54E-3	5.56E-4
Max Contour [mm]	7.59E-4	8.62E-4

over length it can be seen that there are surface marks left by the scallop heights in between passes of the zig-zag toolpath.

From the above experiments, it can be said that the presented trajectory generation algorithm is able to fulfill all of its goals which include redundancy resolution, singularity avoidance, and optimization

6.3 Dual Stage Feed Drive Tracking Control Results

To verify that the dual stage feed drive controller improves tracking error performance, the dual stage feed drive controller is used to follow contours. In this set of experiments, the parameters used for the micromill and rotary table are as shown in Tab. 4.3 and Tab. 4.4.

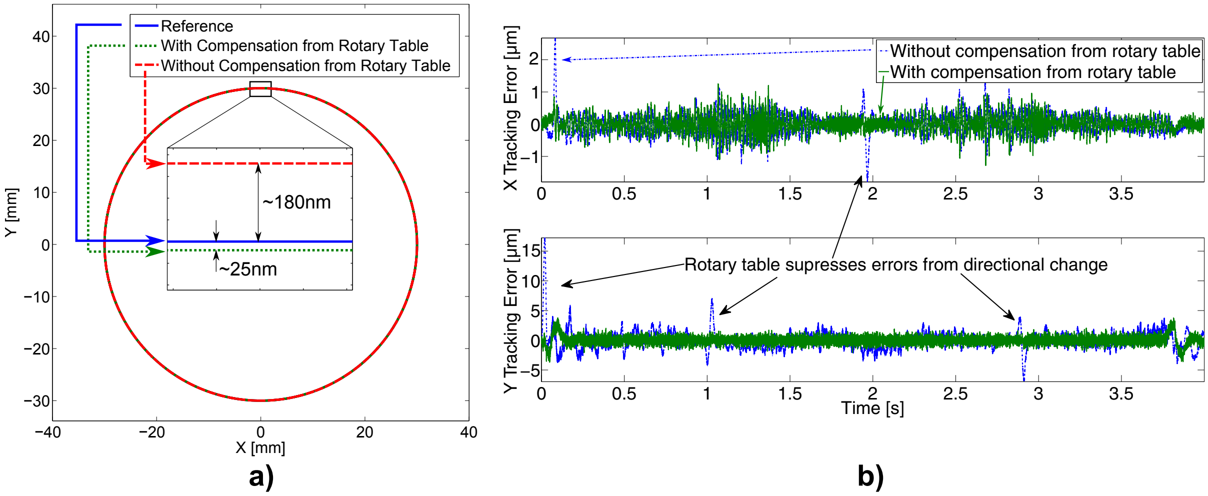


Figure 6.13: a) Circular contour with sliding mode controller and proposed dual stage feed drive control scheme. Radius: 30 mm, Tangential Feed Speed: 600 mm/min. Angular frequency of the traverse : $\omega(Hz) = f/R = 10[Hz]$. b) Tracking error of x and y axis during circular contour

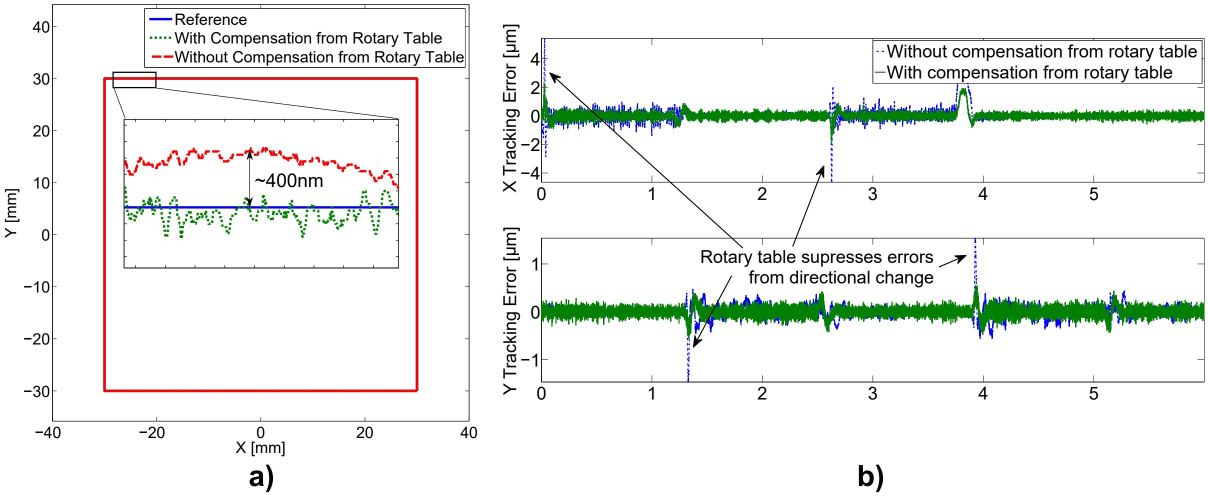


Figure 6.14: a) Square contour with sliding mode controller and proposed dual stage feed drive control scheme. Tangential Feed Speed: 600 mm/min. b) Tracking error of x and y axis during square contour

With these parameters, the micromill has a design bandwidth of 30Hz and the rotary table has a design bandwidth of 250Hz. By having a bandwidth that is approximately 10 times higher than micromill, it is expected that the rotary table will compensate the tracking errors caused by the low bandwidth of the micromill. Two experiments are performed where a circle, as shown in Figure 6.13, and a square, as shown in Figure 6.14, are tracked with the x-y axes of micromachining center. In these experiments x_c and y_c are commanded with the trajectory commands while x_f and y_f compensate for the tracking errors of x_c and y_c respectively. The results can be seen in Fig. 6.13 and Fig. 6.14 and are summarized in Tab. 6.1 and Tab. 6.2, respectively

From Tab. 6.1 and Tab. 6.2, it can be seen that the sliding mode controller is able to keep errors under 3[μm]. The rotary table assists in reducing all tracking errors as shown in Tab. 6.1 and Tab. 6.2. Most notably, the higher bandwidth eliminates friction induced error peaks in the circular contouring where the velocity reversals occur leading to improved contouring performance as shown in Figure 6.13 a) and Figure 6.13 b). Furthermore the dual stage configuration sees improvement in reducing fluctuations around zero tracking error. The fluctuations occur due to quantization noise in the control action. Since these fluctuations occur at frequencies higher than the bandwidth of the rotary table, they are unable to be completely compensated for.

It should be noted that in the case of the square contouring results, as shown in Tab. 6.2, there is a slight increase in contour error. Unlike a circular contour, for a square contour, one of the axis is stationary at all times. While stationary, the axis error will converge to as close to zero as possible since there is an integrator and very low level of disturbance. Since the rotary table has its own oscillatory movement caused by its own feedback noise, this will be added to the stationary coarse actuator resulting in precision that is worse than just the coarse actuator alone.

Table 6.3: Geometric errors with and without compensation

Error	Mean Uncompensated	Mean Compensated	Max Uncompensated	Max Compensated
$d_x\,(x_c)$ [μm]	0.549	0.168	1.07	0.443
$d_y\,(x_c)$ [μm]	0.2	0.0571	0.629	0.265
$d_z\,(x_c)$ [μm]	0.396	0.162	1.1	0.37
$d_x\,(y_c)$ [μm]	1.16	0.222	3.06	0.626
$d_y\,(y_c)$ [μm]	1.65	0.44	4.18	0.883
$d_z\,(y_c)$ [μm]	0.19	0.0791	0.693	0.358
$d_x\,(z_c)$ [μm]	3.91	0.844	7.48	2.4
$d_y\,(z_c)$ [μm]	0.251	0.197	0.9	0.636
$d_z\,(z_c)$ [μm]	0.337	0.115	0.976	0.498

As a result, for a trajectories where the coarse actuator already achieves very high precision on its own, the rotary table may not increase contouring performance. In contrast, for toolpaths where multiple axes are moving there is a performance increase with respect to contouring error as it can be seen from the circular contouring results.

Based on these experiments, it can be concluded that the strength of the dual stage feed drive configuration lies in disturbance rejection for freeform toolpaths. It should be noted that the addition of the rotary table is complementary to any control strategies used on the micromill. Though the sliding mode controller alone has good position tracking properties, the addition of the rotary table increases the overall precision on the respective axis particularly with regards to disturbances. Should a different control strategy, besides the sliding mode controller be used on the micromill, the addition of the rotary table would still be complementary to the overall precision on the respective axis.

6.4 Geometric Error Compensation Results

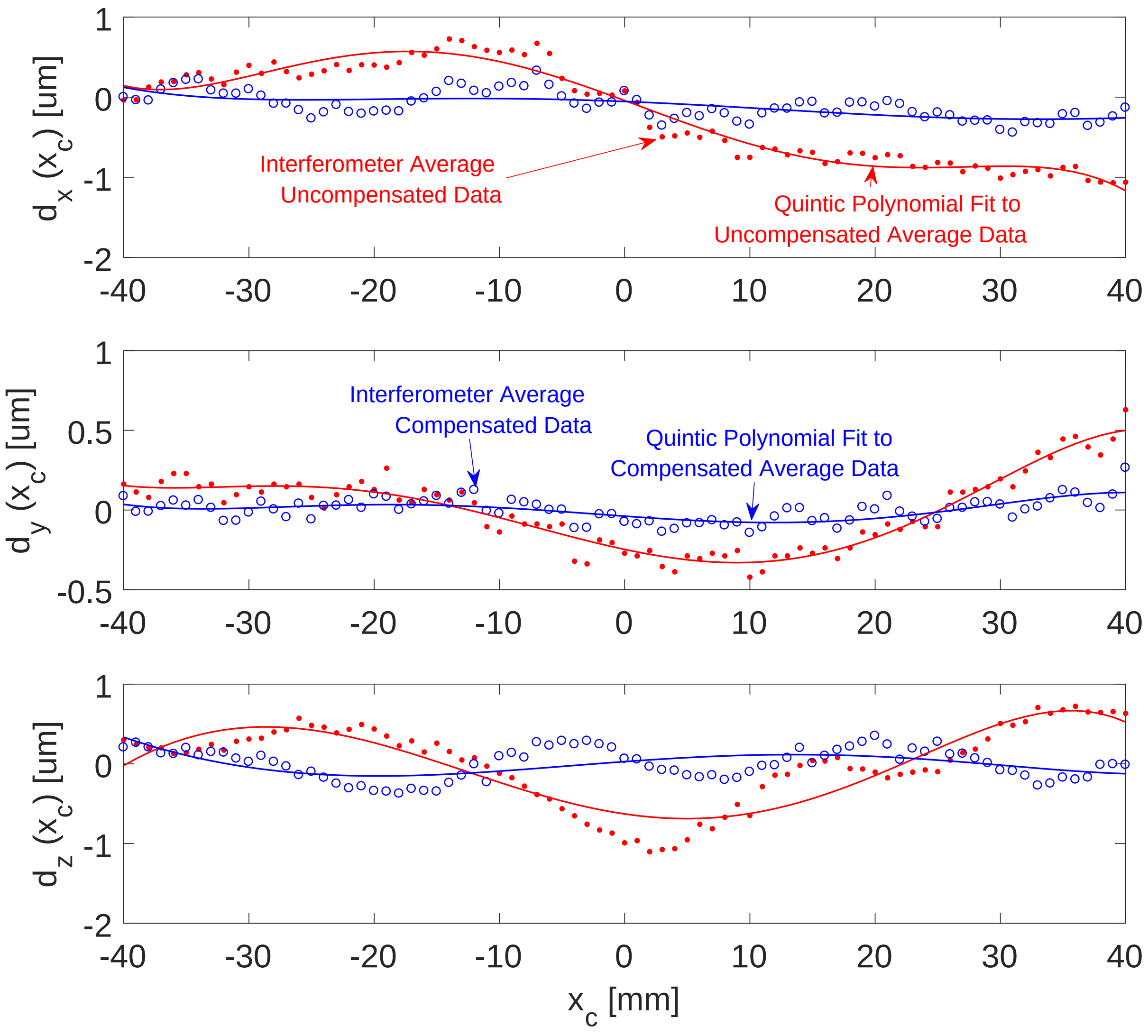


Figure 6.15: Geometric Errors for x_c-axis positions -40mm to 40mm

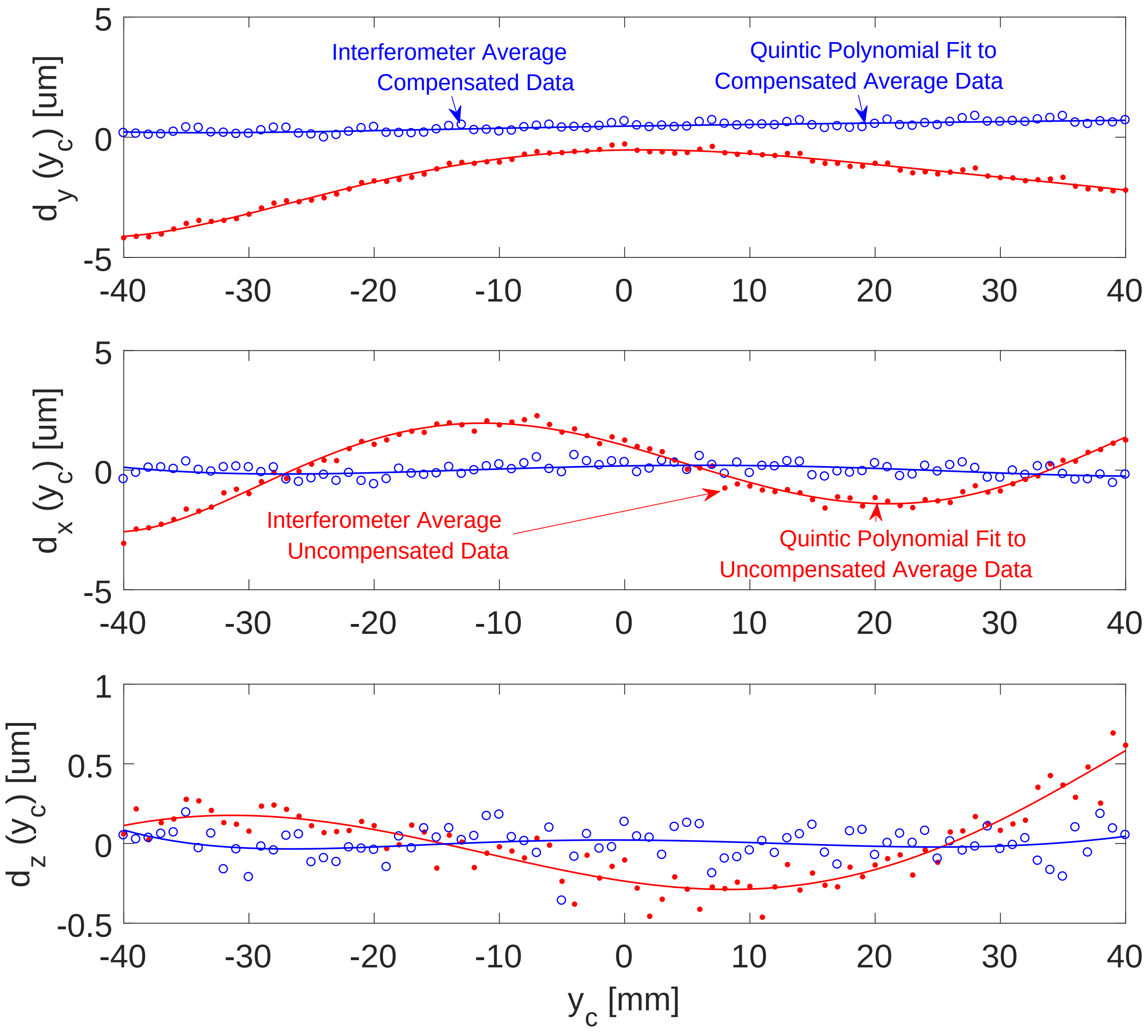


Figure 6.16: Geometric Errors for y_c-axis positions -40mm to 40mm

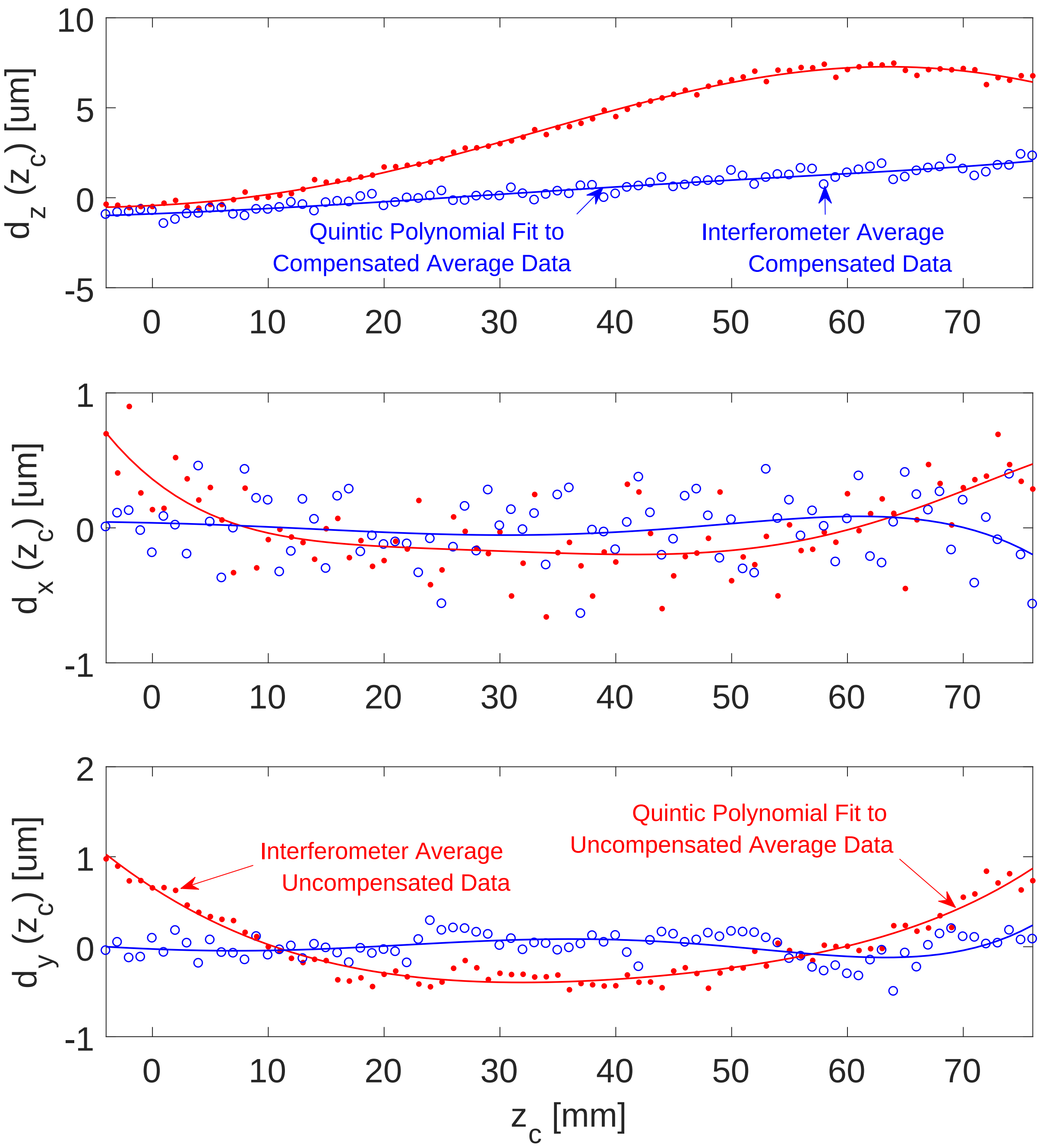


Figure 6.17: Geometric Errors for z_c-axis positions -4mm to 76mm

In addition experiments for the trajectory generation algorithm and dual stage feed drive control, experiments were performed to verify the geometric error compensation algorithm. In these experiments, geometric errors of the 3-axis machine tool are compensated using the high precision stroke of the 6DOF rotary table. First, geometric error measurements were performed with a laser interferometer. It should be noted that due to the inability of the laser interferometer to measure roll errors along an axis, this error was set to 0 in the kinematic model. The measured errors are fit to quintic polynomials then included into the kinematic model, which is used with the gradient descent algorithm to calculate compensating commands for the 6DOF rotary table. To demonstrate the capabilities of the geometric error compensation algorithm, single axis displacement errors are measured with a laser interferometer then compensated with the rotary table with reference commands generated by the algorithm. The objective of these experiments is to demonstrate that positional errors are compensated as the combined effort of multiple axes of the rotary table. The displacement errors, meant to mimic tooltip deviations, are measured with the laser interferometer with and without the compensating action of the rotary table. The results are summarized in Figs. 6.15 to 6.17 and Table 6.3

Based on the results shown in Table 6.3 there is a 64% and 60% improvement in the mean and max geometric error respectively, demonstrating the ability of the algorithm to reduce geometric errors. Furthermore, Figs. 6.15 to 6.17 show consistent improvement of geometric error across the entire actuating range of each of the axis.

To look at the results of the algorithm in greater detail, Fig. 6.18 show the rotary table commands used to compensate for the errors when actuating on the z_c-axis. As it can be seen, the algorithm favors the tilt commands of the rotary table to compensate for the straightness errors in the x and y direction when actuating the z_c-axis. The reason for this is that it is possible to make large displacement corrections with relatively small tilts of the rotary table. Finally, it

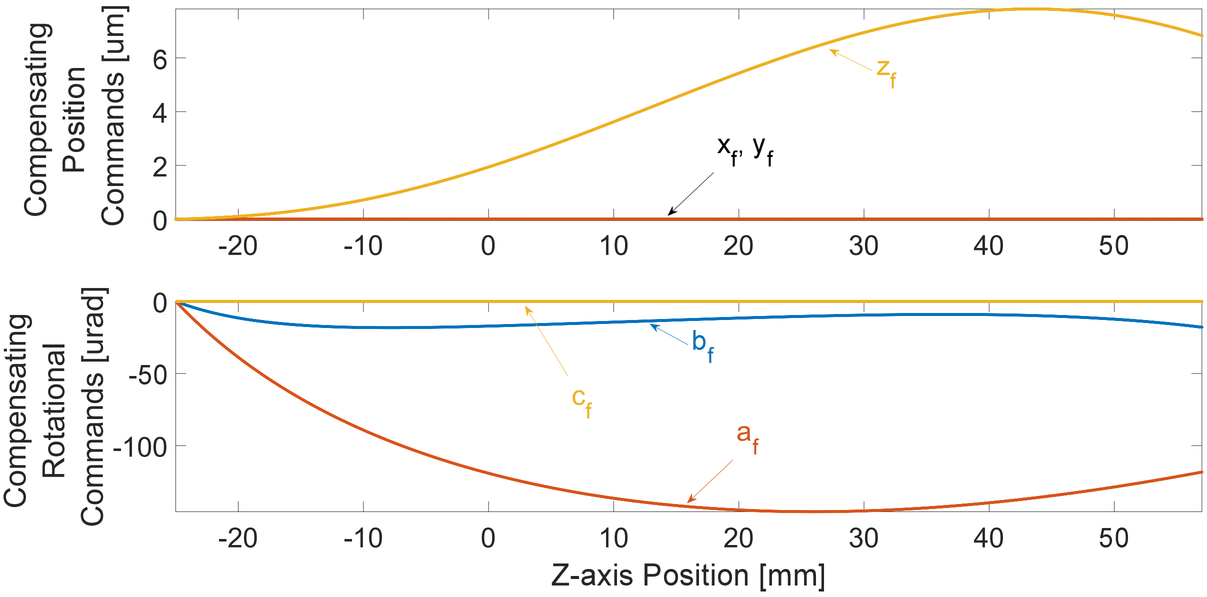


Figure 6.18: Compensating position commands when moving only z_c axis

should be noted that the error is compensated as the combined effort of multiple axes, showing effective use of the compensation Jacobian.

Based upon the single axis experimental results and 2-axis simulation results shown in Fig. 5.3 it can be seen that the tooltip error caused by the geometric errors is relatively small, in the sub-micron range. Due to the inconsistent nature of the cutting process itself, errors originating from the cutting process would be the dominant source of error in the final machined piece. Measurements of a slot cut with a coordinate measuring machine found variations in the error of nearly 30 microns. Furthermore, the configuration of the gantry type milling machine did not allow for the use of a commercial ball bar, which would have allowed measurement of combined errors, independent of process forces.

As an alternative, the full multi-axis compensation capabilities are demonstrated virtually. In order to do this, all 21 geometric errors of a three dimensional toolpath are modeled and the

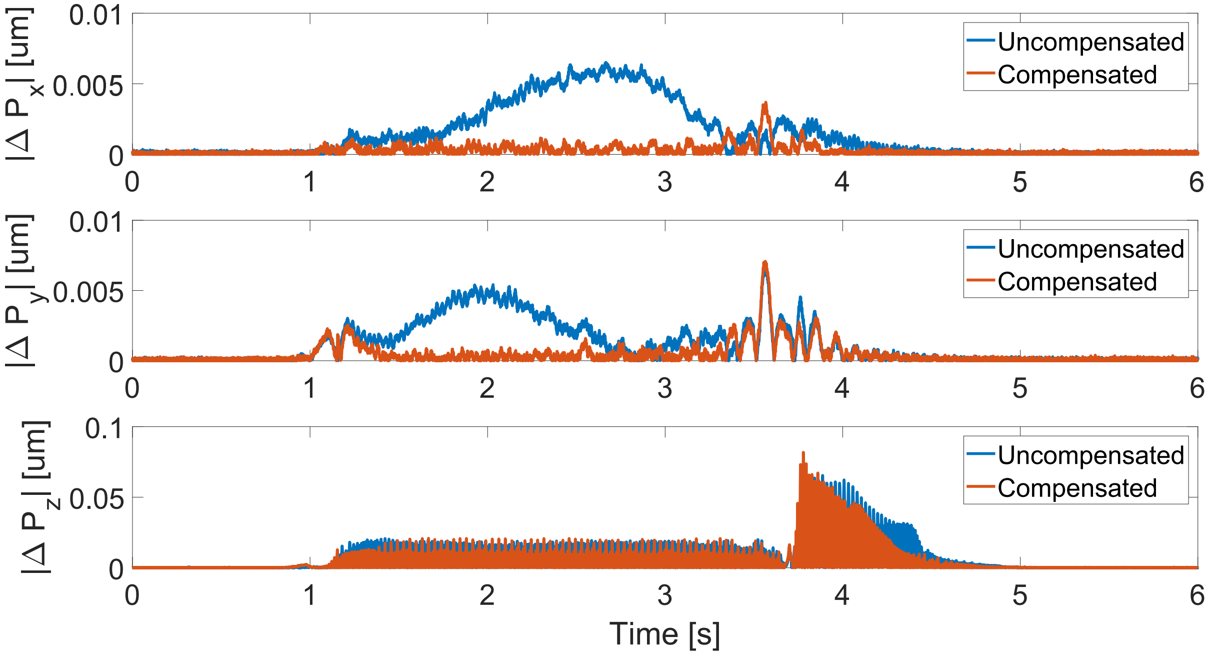


Figure 6.19: Tooltip errors with and without compensation

Table 6.4: Mean and maximum tooltip errors with and without compensation for a multi-axis trajectory

Error	Mean Uncompensated	Mean Compensated	Max Uncompensated	Max Compensated
$\lvert\Delta P_x\rvert$ [μm]	0.97	0.19	6.5	3.7
$\lvert\Delta P_y\rvert$ [μm]	0.79	0.31	6.5	7.1
$\lvert\Delta P_z\rvert$ [μm]	5.2	3.8	65.4	81.8

compensation commands are generated. The compensation commands are then used by the machine as reference commands. The encoder readings are fed into the kinematic model with errors then compared with the ideal kinematic model. In these experiments, a spiral toolpath as shown in Fig. 6.1 is used as the original toolpath for the 3-axis micromill.

Based on the results shown in Tab. 6.4 and Fig. 6.19 there is a 56% and 3% improvement in the mean and max geometric error respectively, demonstrating the ability of the algorithm to reduce geometric errors in multi-axis trajectories. Due to the presence of a tracking error peak at approximately 3.5 seconds this causes the maximum tooltip error be large. Since the compensation algorithm only addresses geometric errors, tracking errors are not accounted for and can appear in the error results. This problem is further demonstrated in the z_c-axis, where there is noisier error behavior than the other axes. This originates from the poor quantization of the control signal of the z_c-axis. Due to the limited number of digital-to-analog converters (DAC) on the dSPACE DS1103, the z_c-axis is controlled with a PWM signal that is converted into voltage signal by a Axiomatic Universal Signal Converter. With the lower resolution of the PWM signal, approximately 9-bits, and further noise and distortion introduced by the universal signal converter, the fluctuations of the tracking error will be higher than the other axes on the machine tool. However, on average, as shown in Tab. 6.4 and Fig. 6.19 the tooltip errors which originate from the geometric errors are compensated for quite well.

Chapter 7

Conclusions

7.1 Conclusions

In this thesis, a trajectory generation algorithm, controls strategy, and geometric error compensation technique have been developed for a novel 9 degree of freedom micromachining center. The hybrid micro-machine tool combines a conventional 3-axis gantry type micromill and a 6DOF high-bandwidth, short stroke magnetic rotary table. Due to its unique 9-axis configuration, new CNC strategies have been developed in the thesis. A 9-axis novel trajectory generation algorithm, which can handle the four redundant drives while respecting the drive limits, have been developed. In order to increase the precision on the translational axes, a control strategy is proposed which combines the high bandwidth and precision of the rotary table and the long stroke of the micromill. Finally, the rotary table is used to compensate for tooltip errors caused by the geometric errors of the machine tool. The contributions are summarized as follows:

The proposed trajectory generation algorithm was developed in order to overcome the challenges associated with generating trajectories for a configuration that has more degrees of freedom than necessary. The position and orientation of a typical cutting tool can be defined by 6 degrees of freedom and achieved by 5 degrees of freedom on a conventional 5-axis CNC machine. Since the developed micro-machine tool has 9 degrees of freedom, 4 more degrees of freedom than necessary, traditional inverse kinematics and feed planning algorithms were not applicable. In this thesis, a methodology was developed to overcome these challenges. A forward kinematic model of the machine tool is first developed, which maps the 9-axis positions to the the tooltip position and tool orientation with respect to the workpiece. A numerical technique is developed, which resolves the redundancies at the differential level with respect to displacement, using the Moore-Penrose inverse of the Jacobian of the forward kinematic model. The proposed differential solution ensures that singularities are avoided and the generated trajectory does not violate the stroke limits of the axes. A corrective 4th Order Runge Kutta numerical integration algorithm is used to extract the position commands from the differential solution, giving axes position commands that correspond to desired tool positions and orientations at fixed displacement intervals along the toolpath. The position commands are then scheduled with respect to time, using a non-linear optimization algorithm to ensure that the toolpath is traversed as fast as possible without violating the velocity, acceleration, and jerk constraints.

In addition to a trajectory generation algorithm, a control strategy was developed which combines the long stroke of the 3-axis micromill with the high bandwidth tracking capabilities of the 6DOF rotary table. Prior to the control design, the rigid body dynamics of the micromill were identified using a linear regression technique. The model is further refined, particularly the non-linear friction characteristics, using disturbance observations with a Kalman filter. Based upon the rigid body model, a sliding mode controller with a bandwidth of approximately 30Hz was designed for position tracking. In order to improve tracking performance, a feedforward friction compensator, based on the LuGre friction model, is implemented. Due to its comparatively simpler dynamics, sophisticated identification was not necessary for the rotary table. Instead, the position controller was designed around a nominal model of the rotary table, which consisted only of a free floating mass. To control position, a lead-lag controller was selected, cascaded with an integrator to remove steady state error. However, it was found that unmodeled flexibilities were limiting the potential bandwidth of the rotary table. Frequency sweeps of the plant found resonance peaks at higher frequencies and notch filters on each axis were implemented, resulting in an achievable bandwidth of approximately 250Hz. In order to combine the long stroke capabilities of the micromill with the high bandwidth tracking capabilities of the rotary table, the tracking error of the micromill is sent as the reference command of the rotary table. Analysis of the transfer function showed that this configuration allowed the axis to adopt the bandwidth of the rotary table. Furthermore, results showed that the rotary table successfully compensated for tracking errors caused by the lower bandwidth of the micromill's three Cartesian drives.

Finally, the rotary table was used to compensate for geometric errors of the machine tool. In order to do this, the effect of the geometric errors of the 3-axis machine tool on the tooltip position had to be modeled. The ideal transformation matrix of each moving axis is augmented to account for the effect of six geometric errors, including positioning error, two straightness errors, roll error, pitch error, and yaw error. The errors are measured with a laser interferometer and fit to a quintic polynomial function of position to preserve jerk continuity, which are then incorporated into the error augmented transformation matrices. The original forward kinematic model is reconstructed with the error augmented transformation matrices, resulting in a tooltip position that accounts for the geometric errors of the 3-axis of the machine tool. By subtracting the result of this forward kinematic model with the original ideal forward kinematic model, it is possible to model the tooltip deviations. Next, rotary table commands are generated to compensate for these tooltip deviations. Due to the non-Cartesian kinematics of rotational errors and the presence of more degrees of freedom than necessary, a non-linear technique was developed in order to generate compensating commands for the rotary table. A gradient descent optimization scheme was developed where the goal was the minimization of tooltip deviations. Since there were more degrees of freedom than necessary, the Moore-Penrose inverse was used again. However, since only the rotary table commands were modified, the Jacobian in this case only accounted for differential changes in tooltip position with respect to changes in rotary table positions. Furthermore, the solution is augmented to ensure the stroke limits of the rotary table were not violated. Single axis laser interferometer experiments showed significant reductions in geometric errors as the combined effort of all 6 axes of the rotary table. Multi-axis simulations results also showed that tooltip deviations were minimized for multi-axis free-form toolpaths.

In summary, the thesis presents a novel 9-axis CNC micro machine tool with a new trajectory generation algorithm, dual axis control algorithm and geometric error compensation strategy. The proposed models can be applied to other multi-axes machine tools with redundant axes.

7.2 Future Research Directions

In regards to the trajectory algorithm presented, further work can be done to select configurations that take advantage of the redundant degrees of freedom. At present, the trajectory generation algorithm only uses the nullspace of the Jacobian to select joint configurations that avoid stroke limits. In reality, it may be possible to achieve multiple goals simultaneously, particularly with so many redundant degrees of freedom. The algorithm presented could be extended to fulfill multiple goals. To name a few, minimization of energy consumption or joint torques can be considered. Specific to the machining process, if it is possible to model the relative stiffness between the tool and the workpiece as a function of the axes positions, then given an analytic gradient, it would be possible to select axes configurations in which the the stiffness is maximized. With the presence of multiple goals, it should be noted that additional strategies need to be developed in prioritizing the goals. With respect to the main redundancy resolution aspect of the trajectory generation algorithm, the Moore-Penrose inverse is used, which consequently minimizes the axes differential with respect to displacement locally. It has been shown in literature that higher order versions of the presented solutions minimize the axes differential at the respective orders. By performing the redundancy resolution at higher order it may be possible to achieve more desirable acceleration and jerk responses when scheduling the commands with respect to time. Finally, globally optimal variations of the presented solution have also been shown in literature and may be an interesting area of research and application for the configuration presented in this work. However it should be noted that due to the globally optimal criteria, even short toolpaths require unrealistic computation time, and optimizing for a real free-form CNC toolpath may be outside the realm of practicality.

Due to its relatively unique configuration, many research directions in the field of controls could be pursued. In a gantry machine tool, one of the lowest frequency structural modes is from the structure of the z-axis itself. The rotary table could be used to provide active damping for this structural mode, increasing the dynamic stiffness between the work piece and the tool. Unlike traditional active damping devices, the rotary table could actively damp structural modes in multiple directions simultaneously. It should be noted that the dual stage feed drive control algorithm presented considers the micromill and rotary table as two separate rigid bodies. This is particularly true for the rotary table acting on the heavier feed drive. In reality, there may be flexibilities that when considered, could increase the precision of the overall system. As a result, state-space control laws which consider the dynamic coupling between the coarse and fine actuator, with the purpose of increasing performance or robustness would be interesting to consider. With respect to geometric error compensation, there would be value in verifying the compensation algorithm for multi-axis trajectories. Though the algorithm has shown effectiveness for single axis experimental results and multi-axis simulation results, verification via a ball-bar would be a possible research direction. Furthermore, the rotary table can also be used for compensation of errors beyond geometric error. If it is possible to model errors caused by thermal expansion or process forces, the tooltip deviations can be modeled. These deviations can be incorporated into the compensation algorithm presented, resulting in rotary table commands that compensate for the above mentioned sources of error. Alternatively, a simpler experimental approach could be taken. A part could be machined and the tooltip deviations could be measured with a CMM. These deviations could be sent to the same compensation algorithm which would generate compensating commands based on CMM measurements. This would require no modeling but a large amount of experimental data.

Finally, in addition to further developing the presented algorithms, further research can be done by applying the presented algorithms to different configurations. Since robotic arms typically have more degrees of freedom than necessary, application of the trajectory generation algorithm presented in this work to a different configuration could be considered novel. Furthermore, since the presented algorithm has an optimization aspect, it would make the most sense to apply this algorithm to robotic milling arms. Likewise, the dual stage feed drive control algorithms can be applied to more coarse/fine actuator configurations found in manufacturing literature. In contrast to using the rotary table as an actuation device, it may be interesting to use the rotary table as a sensing device. If the rotary table is commanded to be held stationary, then the output current to hold the rotary table would be proportional to the cutting force. If the same configuration is used to machine a part, the feedrate of the overall system could be controlled to ensure a constant cutting force is held at all times, using the rotary table as force feedback.

Bibliography

[1] X. Lu, M. Dyck, and Y. Altintas, “Magnetically levitated six degree of freedom rotary table,” *CIRP Annals*, vol. 64, pp. 353–356, Jan. 2015.

[2] M. Dyck, X. Lu, and Y. Altintas, “Magnetically Levitated Rotary Table With Six Degrees of Freedom,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, pp. 530–540, Feb. 2017.

[3] R.-S. Lee and C.-H. She, “Developing a postprocessor for three types of five-axis machine tools,” *The International Journal of Advanced Manufacturing Technology*, vol. 13, pp. 658–665, Sept. 1997.

[4] Psang Dain Lin and Ing Jyh Tsai, “The machining and on-line measurement of spatial cams on four-axis machine tools,” *International Journal of Machine Tools and Manufacture*, vol. 36, pp. 89–101, Jan. 1996.

[5] B. Sencer, Y. Altintas, and E. Croft, “Feed optimization for five-axis CNC machine tools with drive constraints,” *International Journal of Machine Tools and Manufacture*, vol. 48, pp. 733–745, June 2008.

[6] D. E. Whitney, “Resolved Motion Rate Control of Manipulators and Human Prostheses,” *IEEE Transactions on Man-Machine Systems*, vol. 10, pp. 47–53, June 1969.

[7] A. Liegeois, “Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, pp. 868–871, Dec. 1977.

[8] J. Hollerbach and K. Suh, “Redundancy resolution of manipulators through torque optimization,” *IEEE Journal on Robotics and Automation*, vol. 3, pp. 308–316, Aug. 1987.

[9] Y. Halevi, E. Carpanzano, G. Montalbano, and Y. Koren, “Minimum energy control of redundant actuation machine tools,” *CIRP Annals - Manufacturing Technology*, vol. 60, no. 1, pp. 433–436, 2011.

[10] Y. Nakamura and H. Hanafusa, “Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, pp. 163–171, Sept. 1986.

[11] Y. Nakamura and H. Hanafusa, “Optimal Redundancy Control of Robot Manipulators,” *The International Journal of Robotics Research*, vol. 6, pp. 32–42, Mar. 1987.

[12] E. Sanyildiz and H. Temelta, “A comparison study of the numerical integration methods in the trajectory tracking application of redundant robot manipulators,” in *2011 7th International Conference on Electrical and Electronics Engineering (ELECO)*, pp. II–420–II–424, Dec. 2011.

[13] L. Sciavicco and B. Siciliano, “A solution algorithm to the inverse kinematic problem for redundant manipulators,” *IEEE Journal on Robotics and Automation*, vol. 4, pp. 403–410, Aug. 1988.

[14] K. J. Kyriakopoulos and G. N. Saridis, “Minimum jerk path generation,” in *1988 IEEE International Conference on Robotics and Automation Proceedings*, pp. 364–369 vol.1, Apr. 1988.

[15] A. Gasparetto and V. Zanotto, “Optimal trajectory planning for industrial robots,” *Advances in Engineering Software*, vol. 41, pp. 548–556, Apr. 2010.

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

[17] X. Beudaert, S. Lavernhe, and C. Tournier, "Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path," *International Journal of Machine Tools and Manufacture*, vol. 57, pp. 73–82, June 2012.

[18] Y. Altintas and K. Erkorkmaz, "Feedrate optimization for spline interpolation in high speed machine tools," *CIRP Annals*, vol. 52, pp. 297–302, Jan. 2003.

[19] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. Part III: high speed tracking and contouring control of feed drives," *International Journal of Machine Tools and Manufacture*, vol. 41, pp. 1637–1658, Sept. 2001.

[20] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, pp. 65–68, Mar. 1987.

[21] Y. Altintas, K. Erkorkmaz, and W. H. Zhu, "Sliding mode controller design for high speed feed drives," *CIRP Annals*, vol. 49, pp. 265–270, Jan. 2000.

[22] C. Okwudire and Y. Altintas, "Minimum tracking error control of flexible ball screw drives using a discrete-time sliding mode controller," *Journal of Dynamic Systems, Measurement, and Control*, vol. 131, pp. 051006–051006–12, Aug. 2009.

[23] M. Hanifzadegan and R. Nagamune, "Switching gain-scheduled control design for flexible ball-screw drives," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, pp. 014503–014503–6, Sept. 2013.

[24] A. H. H. Hosseinabadi and Y. Altintas, “Modeling and active damping of structural vibrations in machine tools,” *CIRP Journal of Manufacturing Science and Technology*, vol. 7, pp. 246–257, Jan. 2014.

[25] A. Kamalzadeh and K. Erkorkmaz, “Compensation of Axial Vibrations in Ball Screw Drives,” *CIRP Annals*, vol. 56, pp. 373–378, Jan. 2007.

[26] K. Erkorkmaz and A. Kamalzadeh, “High Bandwidth Control of Ball Screw Drives,” *CIRP Annals*, vol. 55, pp. 393–398, Jan. 2006.

[27] A. Kamalzadeh and K. Erkorkmaz, “Accurate tracking controller design for high-speed drives,” *International Journal of Machine Tools and Manufacture*, vol. 47, pp. 1393–1400, July 2007.

[28] S. Staroselsky and K. A. Stelson, “Two-Stage Actuation for Improved Accuracy of Contouring,” in *1988 American Control Conference*, pp. 127–132, June 1988.

[29] O. Masory and J. Wang, “Two-Stage Actuation Control System for CNC Contouring Systems,” in *1992 American Control Conference*, pp. 375–378, June 1992.

[30] A. T. Elfizy, G. M. Bone, and M. A. Elbestawi, “Design and control of a dual-stage feed drive,” *International Journal of Machine Tools and Manufacture*, vol. 45, pp. 153–165, Feb. 2005.

[31] Y. M. Choi and D. G. Gweon, “A High-Precision Dual-Servo Stage Using Halbach Linear Active Magnetic Bearings,” *IEEE/ASME Transactions on Mechatronics*, vol. 16, pp. 925–931, Oct. 2011.

[32] B.-S. Kim, J. Li, and T.-C. Tsao, “Two-parameter robust repetitive control with application to a novel dual-stage actuator for noncircular Machining,” *IEEE/ASME Transactions on Mechatronics*, vol. 9, pp. 644–652, Dec. 2004.

[33] M. Kobayashi and R. Horowitz, “Track seek control for hard disk dual-stage servo systems,” *IEEE Transactions on Magnetics*, vol. 37, pp. 949–954, Mar. 2001.

[34] M. Kobayashi, S. Nakagawa, and S. Nakamura, “A phase-stabilized servo controller for dual-stage actuators in hard disk drives,” *IEEE Transactions on Magnetics*, vol. 39, pp. 844–850, Mar. 2003.

[35] G. Herrmann, M. C. Turner, I. Postlethwaite, and G. Guo, “Practical implementation of a novel anti-windup scheme in a HDD-dual-stage servo-system,” *IEEE/ASME Transactions on Mechatronics*, vol. 9, pp. 580–592, Sept. 2004.

[36] W. H. Yao and M. Tomizuka, “Robust controller design for a dual-stage positioning system,” in *Proceedings of the IECON ’93 International Conference on Industrial Electronics, Control, and Instrumentation*, pp. 62–66 vol.1, Nov. 1993.

[37] T.-L. Tai and J.-S. Chen, “Discrete-time sliding-mode controller for dual-stage systemsA hierarchical approach,” *Mechatronics*, vol. 15, pp. 949–967, Oct. 2005.

[38] J. H. She, X. Xin, and Y. Pan, “Equivalent-Input-Disturbance Approach-Analysis and Application to Disturbance Rejection in Dual-Stage Feed Drive Control System,” *IEEE/ASME Transactions on Mechatronics*, vol. 16, pp. 330–340, Apr. 2011.

[39] J. Zheng and M. Fu, “Nonlinear Feedback Control of a Dual-Stage Actuator System for Reduced Settling Time,” *IEEE Transactions on Control Systems Technology*, vol. 16, pp. 717–725, July 2008.

[40] G. Zhang, R. Veale, T. Charlton, B. Borchardt, and R. Hocken, “Error Compensation of Coordinate Measuring Machines,” *CIRP Annals*, vol. 34, pp. 445–448, Jan. 1985.

[41] A. C. Okafor and Y. M. Ertekin, “Derivation of machine tool error models and error compensation procedure for three axes vertical machining center using rigid body kinematics,” *International Journal of Machine Tools and Manufacture*, vol. 40, pp. 1199–1213, June 2000.

[42] W. T. Lei and Y. Y. Hsu, “Accuracy enhancement of five-axis CNC machines through real-time error compensation,” *International Journal of Machine Tools and Manufacture*, vol. 43, pp. 871–877, July 2003.

[43] N. Huang, Y. Jin, Q. Bi, and Y. Wang, “Integrated post-processor for 5-axis machine tools with geometric errors compensation,” *International Journal of Machine Tools and Manufacture*, vol. 94, pp. 65–73, July 2015.

[44] Y. Y. Hsu and S. S. Wang, “A new compensation method for geometry errors of five-axis machine tools,” *International Journal of Machine Tools and Manufacture*, vol. 47, pp. 352–360, Feb. 2007.

[45] S. Zhu, G. Ding, S. Qin, J. Lei, L. Zhuang, and K. Yan, “Integrated geometric error modeling, identification and compensation of CNC machine tools,” *International Journal of Machine Tools and Manufacture*, vol. 52, pp. 24–29, Jan. 2012.

[46] M. Tsutsumi, S. Tone, N. Kato, and R. Sato, “Enhancement of geometric accuracy of five-axis machining centers based on identification and compensation of geometric deviations,” *International Journal of Machine Tools and Manufacture*, vol. 68, pp. 11–20, May 2013.

[47] S. Xiang and Y. Altintas, “Modeling and compensation of volumetric errors for five-axis machine tools,” *International Journal of Machine Tools and Manufacture*, vol. 101, pp. 65–78, Feb. 2016.

[48] O.-S. Kim, S.-H. Lee, and D.-C. Han, “Positioning performance and straightness error compensation of the magnetic levitation stage supported by the linear magnetic bearing,” *IEEE Transactions on Industrial Electronics*, vol. 50, pp. 374–378, Apr. 2003.

[49] Y. Deng, X. Jin, and Z. Zhang, “A macromicro compensation method for straightness motion error and positioning error of an improved linear stage,” *The International Journal of Advanced Manufacturing Technology*, vol. 80, pp. 1799–1806, Oct. 2015.

[50] R. P. Paul, *Robot manipulators: mathematics, programming, and control : the computer control of robot manipulators*. MIT Press, 1981. Google-Books-ID: m99TAAAAMAAJ.

[51] A. Yuen, K. Zhang, and Y. Altintas, “Smooth trajectory generation for five-axis machine tools,” *International Journal of Machine Tools and Manufacture*, vol. 71, pp. 11–19, Aug. 2013.

[52] M. Heng and K. Erkorkmaz, “Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability,” *International Journal of Machine Tools and Manufacture*, vol. 50, pp. 281–293, Mar. 2010.

[53] L. Piegl and W. Tiller, *The NURBS Book*. Monographs in Visual Communication, Berlin Heidelberg: Springer-Verlag, 1995.

[54] K. Erkorkmaz and Y. Altintas, “High speed CNC system design. Part II: modeling and identification of feed drives,” *International Journal of Machine Tools and Manufacture*, vol. 41, pp. 1487–1509, Aug. 2001.

[55] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, pp. 35–45, Mar. 1960.

[56] V. I. Utkin, “Sliding mode control design principles and applications to electric drives,” *IEEE Transactions on Industrial Electronics*, vol. 40, pp. 23–36, Feb. 1993.

[57] C. C. De Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, “A new model for control of systems with friction,” *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pp. 419–425, 1995.

[58] P. I. Ro, W. Shim, and S. Jeong, “Robust friction compensation for submicrometer positioning and tracking for a ball-screw-driven slide system,” *Precision Engineering*, vol. 24, no. 2, pp. 160–173, 2000.

[59] M. Dyck, “Magnetically levitated six degree of freedom micro-machining rotary table,” Master’s thesis, University of British Columbia, 2014.

Appendix A

Foward Kinematic Equation with Geometric Errors

The actual tooltip position, $\mathbf{P}_a\left(\mathbf{q}\right) = \left[P_{x,a}\left(\mathbf{q}\right), P_{y,a}\left(\mathbf{q}\right), P_{z,a}\left(\mathbf{q}\right)\right]^T$ are shown detail in the following equations:

$$
\begin{aligned}
P_{x,a}\left(\mathbf{q}\right) = \; & -\varepsilon_x(x_c)((L + L_{x,z})(\mathsf{c}_{a_f}\mathsf{s}_{c_f} + \mathsf{c}_{c_f}\mathsf{s}_{a_f}\mathsf{s}_{b_f}) \\
& +(L_{sp,y} - y_c)(\mathsf{s}_{a_f}\mathsf{s}_{c_f} - \mathsf{c}_{a_f}\mathsf{c}_{c_f}\mathsf{s}_{b_f})) \\
& +\varepsilon_x(z_c)((L_{sp,z} + L_t)(\mathsf{c}_{a_f}\mathsf{s}_{c_f} + \mathsf{c}_{c_f}\mathsf{s}_{a_f}\mathsf{s}_{b_f}) \\
& +L_{sp,y}(\mathsf{s}_{a_f}\mathsf{s}_{c_f} - \mathsf{c}_{a_f}\mathsf{c}_{c_f}\mathsf{s}_{b_f})) \\
& +\varepsilon_y(x_c)((L + L_{x,z})\mathsf{c}_{b_f}\mathsf{c}_{c_f} + x_c(\mathsf{c}_{a_f}\mathsf{c}_{c_f}\mathsf{s}_{b_f} - \mathsf{s}_{a_f}\mathsf{s}_{c_f})) \\
& +\varepsilon_y(y_c)((L)\mathsf{c}_{b_f}\mathsf{c}_{c_f}) \\
& +\varepsilon_z(x_c)((L_{sp,y} - y_c)\mathsf{c}_{b_f}\mathsf{c}_{c_f} + x_c(\mathsf{c}_{a_f}\mathsf{s}_{c_f} + \mathsf{c}_{c_f}\mathsf{s}_{a_f}\mathsf{s}_{b_f})) \\
& -\varepsilon_y(z_c)((L_{sp,z} + L_t)\mathsf{c}_{b_f}\mathsf{c}_{c_f}) \\
& -\varepsilon_x(y_c)(L(\mathsf{c}_{a_f}\mathsf{s}_{c_f} + \mathsf{c}_{c_f}\mathsf{s}_{a_f}\mathsf{s}_{b_f}) \\
& +(L_{sp,y} - y_c)(\mathsf{s}_{a_f}\mathsf{s}_{c_f} - \mathsf{c}_{a_f}\mathsf{c}_{c_f}\mathsf{s}_{b_f})) \\
& +\varepsilon_z(y_c)((L_{sp,y} - y_c)\mathsf{c}_{b_f}\mathsf{c}_{c_f}) \\
& +(-\delta_y(x_c) - \delta_y(y_c) + \delta_y(z_c))(\mathsf{c}_{a_f}\mathsf{s}_{c_f} + \mathsf{c}_{c_f}\mathsf{s}_{a_f}\mathsf{s}_{b_f}) \\
& +(-\delta_z(x_c) - \delta_z(y_c) + \delta_z(z_c))(\mathsf{s}_{a_f}\mathsf{s}_{c_f} - \mathsf{c}_{a_f}\mathsf{c}_{c_f}\mathsf{s}_{b_f}) \\
& +(L_{sp,y} - y_c - y_f)\mathsf{c}_{a_f}\mathsf{s}_{c_f} \\
& +(-\delta_x(x_c) - \delta_x(y_c) + \delta_x(z_c) \\
& -x_c - x_f - L_{sp,y}\varepsilon_z(z_c))\mathsf{c}_{b_f}\mathsf{c}_{c_f} \\
& +(-L - L_s - L_{x,z} - z_f - z_{f,0})\mathsf{s}_{a_f}\mathsf{s}_{c_f} \\
& +(L + L_s + L_{x,z} + z_f + z_{f,0})\mathsf{c}_{a_f}\mathsf{c}_{c_f}\mathsf{s}_{b_f} \\
& +(L_{sp,y} - y_c - y_f)\mathsf{c}_{c_f}\mathsf{s}_{a_f}\mathsf{s}_{b_f} \\
& -\gamma_{x,z_c}((-L + L_{y,z})(\mathsf{c}_{a_f}\mathsf{s}_{c_f} + \mathsf{c}_{c_f}\mathsf{s}_{a_f}\mathsf{s}_{b_f}) \\
& +L_{sp,y}(\mathsf{c}_{a_f}\mathsf{c}_{c_f}\mathsf{s}_{b_f} - \mathsf{s}_{a_f}\mathsf{s}_{c_f})) \\
& +\gamma_{y,z_c}((-L + L_{y,z})\mathsf{c}_{b_f}\mathsf{c}_{c_f}) \\
& +\gamma_{z,x_c}((L_{sp,y} - y_c)\mathsf{c}_{b_f}\mathsf{c}_{c_f}) \\
& +\mathcal{O}(\varepsilon^2, \delta^2, \varepsilon\delta)
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
P_{y,a}\left(\mathbf{q}\right) \;=\;& \varepsilon_x(x_c)((L+L_{x,z})(\mathsf{s}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f}-\mathsf{c}_{a_f}\mathsf{c}_{c_f})\\
&+(-L_{sp,y}+y_c)(\mathsf{c}_{c_f}\mathsf{s}_{a_f}+\mathsf{c}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f}))\\
&+\varepsilon_x(z_c)((L_{sp,z}+L_t)(\mathsf{c}_{a_f}\mathsf{c}_{c_f}-\mathsf{s}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f})\\
&+L_{sp,y}(\mathsf{c}_{c_f}\mathsf{s}_{a_f}+\mathsf{c}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f}))\\
&+\varepsilon_x(y_c)(L(\mathsf{s}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f}-\mathsf{c}_{a_f}\mathsf{c}_{c_f})\\
&+(-L_{sp,y}+y_c)(\mathsf{c}_{c_f}\mathsf{s}_{a_f}+\mathsf{c}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f}))\\
&-\varepsilon_z(x_c)((L_{sp,y}-y_c)\mathsf{c}_{b_f}\mathsf{s}_{c_f}+x_c(\mathsf{s}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f}-\mathsf{c}_{a_f}\mathsf{c}_{c_f}))\\
&+\varepsilon_y(z_c)((L_{sp,z}+L_t)\mathsf{c}_{b_f}\mathsf{s}_{c_f})\\
&-\varepsilon_y(y_c)(L\mathsf{c}_{b_f}\mathsf{s}_{c_f})\\
&-\varepsilon_y(x_c)((L+L_{x,z})\mathsf{c}_{b_f}\mathsf{s}_{c_f}+x_c(\mathsf{c}_{c_f}\mathsf{s}_{a_f}+\mathsf{c}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f}))\\
&-\varepsilon_z(y_c)((L_{sp,y}-y_c)\mathsf{c}_{b_f}\mathsf{s}_{c_f})\\
&+(-\delta_y(x_c)-\delta_y(y_c)+\delta_y(z_c))(\mathsf{c}_{a_f}\mathsf{c}_{c_f}-\mathsf{s}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f})\\
&+(-\delta_z(x_c)-\delta_z(y_c)+\delta_z(z_c))(\mathsf{c}_{c_f}\mathsf{s}_{a_f}+\mathsf{c}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f})\\
&+(L_{sp,y}-y_c-y_f)\mathsf{c}_{a_f}\mathsf{c}_{c_f}\\
&+(-L-L_s-L_{x,z}-z_f-z_{f,0})\mathsf{c}_{c_f}\mathsf{s}_{a_f}\\
&+(\delta_x(x_c)+\delta_x(y_c)-\delta_x(z_c)\\
&+x_c+xf+L_{sp,y}\varepsilon_z(z_c))\mathsf{c}_{b_f}\mathsf{s}_{c_f}\\
&+(-L-L_s-L_{x,z}-z_f-z_{f,0})\mathsf{c}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f}\\
&+(-L_{sp,y}+y_c+y_f)\mathsf{s}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f}\\
&-\gamma_{z,x_c}((L_{sp,y}-y_c)\mathsf{c}_{b_f}\mathsf{s}_{c_f})\\
&-\gamma_{y,z_c}((-L+L_{y,z})\mathsf{c}_{b_f}\mathsf{s}_{c_f})\\
&+\gamma_{x,z_c}((L-L_{y,z})(\mathsf{c}_{a_f}\mathsf{c}_{c_f}-\mathsf{s}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f})\\
&+L_{sp,y}(\mathsf{c}_{c_f}\mathsf{s}_{a_f}+\mathsf{c}_{a_f}\mathsf{s}_{b_f}\mathsf{s}_{c_f}))\\
&+\mathcal{O}(\varepsilon^2,\delta^2,\varepsilon\delta)
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
P_{z,a}\left(\mathbf{q}\right) \;=\;& \varepsilon_x(y_c)(L\mathsf{c}_{b_f}\mathsf{s}_{a_f} + (-L_{sp,y} + y_c)\mathsf{c}_{a_f}\mathsf{c}_{b_f}) \\
& -L_w - \varepsilon_y(z_c)(L_{sp,z}\mathsf{s}_{b_f} + L_t\mathsf{s}_{b_f}) \\
& -L_r + \varepsilon_z(y_c)(L_{sp,y}\mathsf{s}_{b_f} - y_c\mathsf{s}_{b_f}) \\
& +\varepsilon_x(x_c)((L + L_{x,z})\mathsf{c}_{b_f}\mathsf{s}_{a_f} + (-L_{sp,y} + y_c)\mathsf{c}_{a_f}\mathsf{c}_{b_f}) \\
& +(-\delta_x(x_c) - \delta_x(y_c) + \delta_x(z_c) - L_{sp,y}\varepsilon_z(z_c))\mathsf{s}_{b_f} \\
& +(-x_c - x_f)\mathsf{s}_{b_f} \\
& +\varepsilon_y(x_c)((L + L_{x,z})\mathsf{s}_{b_f} - x_c\mathsf{c}_{a_f}\mathsf{c}_{b_f}) \\
& -\varepsilon_z(x_c)((y_c - L_{sp,y})\mathsf{s}_{b_f} + x_c\mathsf{c}_{b_f}\mathsf{s}_{a_f}) \\
& +\varepsilon_y(y_c)((L)\mathsf{s}_{b_f}) \\
& -\varepsilon_x(z_c)((L_{sp,z} + L_t)\mathsf{c}_{b_f}\mathsf{s}_{a_f} - L_{sp,y}\mathsf{c}_{a_f}\mathsf{c}_{b_f}) \\
& +(-L - L_s - L_{x,z} - \delta_z(x_c) - \delta_z(y_c) \\
& +\delta_z(z_c) - z_f - z_{f,0})\mathsf{c}_{a_f}\mathsf{c}_{b_f} \\
& +(-L_{sp,y} + \delta_y(x_c) + \delta_y(y_c) \\
& -\delta_y(z_c) + y_c + y_f)\mathsf{c}_{b_f}\mathsf{s}_{a_f} \\
& +\gamma_{z,x_c}((L_{sp,y} - y_c)\mathsf{s}_{b_f}) \\
& +\gamma_{y,z_c}((-L + L_{y,z})\mathsf{s}_{b_f}) \\
& +\gamma_{x,z_c}(L_{sp,y}\mathsf{c}_{a_f}\mathsf{c}_{b_f} + (-L + L_{y,z})\mathsf{c}_{b_f}\mathsf{s}_{a_f}) \\
& +\mathcal{O}(\varepsilon^2, \delta^2, \varepsilon\delta)
\end{aligned}
\tag{A.3}
$$

where $L = L_{sp,z} - L_{0,z} + L_t + L_{x,z} - z_c$ and $\mathcal{O}(\varepsilon^2, \delta^2, \varepsilon\delta)$ are the second order and higher order error terms.