Machining Speed Gains in a 3-Axis CNC Lathe Mill

by

James Rigsby

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2010

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

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

Abstract

The intent of this work is to improve the machining speed of an existing 3 axis CNC wood working lathe. This lathe is unique in that it is a modified manual lathe that is capable of machining complex sculptured surfaces. The current machining is too slow for the lathe to be considered useful in an industrial setting. To improve the machining speed of the lathe, several modifications are made to the mechanical, electrical and software aspects of the system.

It was found that the x-axis of the system, the axis that controls the depth of cut of the tool, is the limiting axis. A servo motor is used to replace the existing stepper motor, providing the x-axis with more torque and faster response times, which should improve the performance of the system. To control the servo motor, a 1st-order linear transfer function model is selected and identified. Then, an adaptive sliding mode controller is applied to make the x-axis a robust and accurate positioning system. A new trajectory generator is implemented to create a smooth motion for all three axes of the lathe. This trajectory uses a 5th-order polynomial to describe the position curve of the feed profile, giving the system continuous jerk motion. This type of profile is much easier for motors to follow, as discontinuous motion will always result in errors. These modifications to the lathe system are then evaluated experimentally using a test case. Three test pieces are designed to represent three of the common shapes that are typically machined on the wood turning lathe. These test cases indicated a minimum reduction in machining time of 52.91% over the previous lathe system.

An algorithm is also developed that attempts to sacrifice work piece model geometry to achieve speed gains. The algorithm is used when a certain feedrate is desired for a model, but machining at that speed will cause toolpath following errors, leaving surface defects in the work piece. The algorithm will attempt to solve this problem by sacrificing model geometry. A simulation tool is used to detect where surface defects will occur during machining and a then the work piece model is modified in the corresponding area. This will create a smoother part, which allows each axis of the system to follow the new toolpath more easily, as the dynamic requirements are reduced. The potential of this algorithm is demonstrated in an experimental test case. A test piece is created that has features of varying difficulty to machine. When the algorithm is run, Matlab/Simulink is used simulate the output of the lathe and locate the areas in the part geometry that will

cause defects. Once located, the geometry features are smoothed in SolidWorks using the fillet feature. The algorithm produces a work piece with smoothed geometry that can be machined at a feedrate approximately 42.8% faster than before. Although it is only the first implementation of the algorithm, the experimental results confirm the potential of the method. Machining speed gains are successfully achieved through the sacrifice of model geometry.

Acknowledgements

I would like to thank both Professor Bedi and Professor Hulls for all of their guidance and support throughout this thesis. I will be able to use the knowledge and experience they shared with me for years to come.

I would like to acknowledge Gerardo Salas for providing the SolidWorks models of the CNC lathe. I would also like to thank both Kandarp Patel and Gerardo Salas for their work on creating the toolpath generator used in this work. And finally I would like to thank Arun Das for his help in implementing the geometry smoothing algorithm.

Thank you all.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The lathe has been used in industrial applications for several centuries [27]. Within that time frame there have been many improvements to the individual parts that make up a lathe. These improvements range from the durability of materials to the rotation speeds possible, as well as the ease of use of the system as a whole [15]. More recently, within the last 50 years, computers have begun to play a larger role in the operation of a lathe in the form of computer numerical control (CNC). Modern CNC lathe systems are capable of following preprogrammed paths accurately and at high speeds. However, for the most part these systems, like their modern manual lathe counterparts, are still only able to machine work pieces that are axis-symmetrical. Although a great deal of research has gone into improving the operation of the lathe, the basic product being machined has remained the same over the years. Some CNC lathes are able to make non-symmetric parts through the use of three controlled axes, which includes control of the axis of rotation and two degrees of control for the tool. Although the degrees of freedom are present, the parts that are machined with these systems are typically simple pieces that can have a high throughput rate to justify the high cost of the machines. Although some research has been conducted in the field of making non-symmetric shapes on lathes [11], the majority of these types of shapes in industry are still made on 3- and 5- axis milling centres.

1.2 Current Research

A great deal of research has been conducted over the last 8 years in the Surface Machining group at the University of Waterloo in the area of complex sculptured surfaces. These surfaces, which are defined in detail later in this work, are essentially complicated and even artistic work pieces that have typically only been achievable through manual hand carving up until the last few years. In particular, research has been conducted in creating 3D sculptured surfaces using wood turning lathes as the machining centre. Many different lathe configurations and control strategies have been considered to best suit the machining of a sculptured surface. There also exists a large disconnect with most CNC systems between the development of a work piece model and creating the actual part. Quite often in industry, uniquely skilled workers are required to design the part models, create the necessary toolpaths and then program and run the machining system. These steps are often assisted with CAD/CAM packages, but still require a large amount of time and knowledge. A great deal of work has been done at the Surface Machining group to bridge this disconnect. Primarily, several software packages have been created for the direct conversion of a solid model into a toolpath that is usable by a machining centre. One goal of these works has been to make the machining process simpler and more intuitive; not just for industry, but also for the general consumer. The idea is to develop an affordable wood turning lathe that can create complex sculptured surfaces, with intuitive software that is run on a PC, from someone's home.

1.3 Earlier Works

Prior to the research presented in this work, there were several iterations of CNC lathe systems that were capable of machining complex sculptured surfaces. The first was created by Manos [15]. He constructed a simple low cost lathe that would be suitable for a general consumer. This work was used to demonstrate the ability of a lathe to machine a complex sculptured surface and did so successfully. The system also performed machining by only controlling a single axis with a computer and motor, while the other two axes were controlled by human input. This setup was used to reduce the cost of the system, but created drawbacks in the form of machining accuracy that compromised the surface finish of the work pieces. This lathe system also implemented a software technique called 'dropping the

ball', which is described in detail in section 6.1, that created toolpaths directly from work piece models.

As an extension to this work, Poon [21] developed a CNC controller architecture that condensed the multiple steps of regular CNC controllers into one design. This architecture was tested on the same wood turning lathe developed by Manos [15]. Modifications were made to the system that added a second motor to give computer control to two of the systems axes, which improved the speed of the system along with the accuracy. The results encouraged the further development of autonomous control algorithms for CNC lathe systems.

1.4 Recent Work

Having successfully proven the concept of machining complex sculptured surfaces on a CNC lathe, the research focus then shifted towards improving the quality and efficiency of the machining process. A new lathe structure was prototyped using an existing manual lathe design. Although larger and more expensive, the system was still significantly cheaper than a typical CNC lathe system. The intent was to create a system that could machine complex sculptured surfaces at a speeds faster than those achieved with the previous lathe designs while also improving the surface quality of the part and keeping the cost of the system low. This type of system would be ideal for small businesses, which would benefit from a CNC machining system that used a simplified control architecture. The main test pieces that were machined on this lathe were table legs. Table legs are one example of an item that is machined commercially on lathes. However, most table legs consist of axis-symmetric designs. Those that do not are typically hand carved. For this reason it was thought that machining table legs consisting of complex sculptured surface designs would be the perfect industrial application for demonstrating the abilities of the unique 3-axis CNC lathe. Below in figure 1.1 are a few illustrations of the wide variety of table legs that were machined with this CNC lathe system.

Figure 1.1: Example Table Leg Designs

1.5 Brief System Description

The original lathe used stepper motors on all three axes to provide complete computerized control over the system. Figure 1.2 illustrates the motion capabilities of each axis. As can be seen in the diagram, the work piece rotates on the rotational axis, labeled the c-axis. The cutting tool is given two degrees of freedom. One degree comes from the translational, or z-axis; which moves the tool along the length of the work piece. The second comes from the radial, or x-axis; which controls the radial depth of cut of the tool into the work piece. However, the use of stepper motors was found to limit the dynamic capabilities of the system. The motors were not able to produce the necessary torques to follow the complex sculptured surfaces at the desired speeds. This caused the motors to slip and create defects in the surfaces of the work pieces. This problem was compounded by the method used for creating trajectory profiles from the work piece toolpaths for the motors to follow. The system employed a linear interpolation method between the toolpath points. This created trajectories that were difficult for the motors to follow, as they called for discontinuous velocity and acceleration movements, which are impossible for the motors to follow. This inevitably led to the stepper motors missing more counts and creating more defects in the surface finish of a work piece when machining at faster speeds.

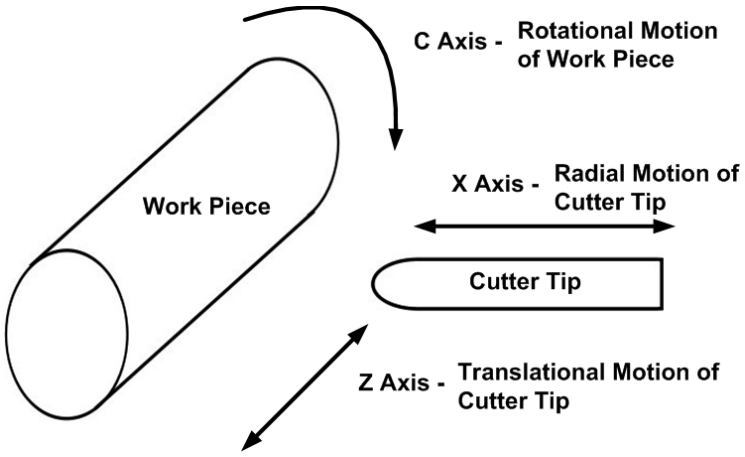


Figure 1.2: Lathe Degrees of Motion

1.6 Project Motivations

The limitations that were discovered in the most recent 3-axis CNC lathe design were the main motivators for the work presented in this thesis. As mentioned, the intention of the previous lathe design was to create a system that was able to machine complex sculptured surfaces precisely and use a simplified control architecture. However, due to the limitations in the system dynamics that were discussed, although capable of machining complex sculptured surfaces, these pieces required a slow machining speed to produce acceptable results. As was demonstrated by machining sculptured table legs, the finished products were exquisite and quite unique by industry standards, but the time required to machine these products was longer than acceptable. Because of the promise that was shown with this technology, it was decided to try and further improve the performance of this system.

1.7 Thesis Concept

This thesis attempts to improve the performance of the 3-axis CNC lathe system by increasing its machining speed. This will be accomplished by first modifying the mechanical and electrical structure of the lathe by replacing the stepper motor of the x-axis with a servo motor because it was found to be the speed limiting axis of the system. A theoretical model will then be developed for the x-axis and a robust and accurate control law will be implemented. The method of trajectory interpolation will be reviewed to find an alternative method that will provide a smoother profile that is easier for the motors of the system to follow. Finally, an algorithm will be explored that attempts to further improve the machining speed of the lathe system by sacrificing model geometry.

1.8 Thesis Layout

The remainder of this thesis has the following layout: Chapter 2 reviews the relevant literature for the application of using a 3-axis lathe for machining complex sculptured surfaces and the modifications that will be applied to improve its performance. A detailed description of how the lathe operates and of the physical modifications that were made to

its structure is provided in Chapter 3. The development of the model and controller for the new servo motor added to the lathe are presented in chapters 4 and 5 respectively. The algorithms used to operate the CNC lathe, namely the trajectory algorithm and the geometry sacrificing algorithm, are described in Chapter 6. Simulated and experimental test results are then presented in Chapter 7, exploring what improvements have been achieved in the machining speed of the system. The experimental results present two test cases; one for evaluating the performance of the lathe machining some basic shapes and one to evaluate the geometry sacrificing algorithm specifically. Finally, the concluding remarks and future works are discussed in Chapter 8.

Chapter 2

Literature Review

As just described in the previous section, the 3-axis lathe is presently able to machine complex sculptured surfaces. However, due to several factors, namely the motor performance and trajectory planning strategies, the machining time of these surfaces is quite slow. For a complex surface, the rotational axis of the lathe only rotates at approximately $5 - 10rpm$. For a full length table leg, this rotation speed translates into a machining time of approximately 90 minutes. This section will discuss the areas being investigated for improving the machining speed of the 3-axis lathe. The current and previous research in this field will be outlined to provide context for the choices that have been made in this work.

2.1 3-axis Lathe Technology

In industry, the CNC lathe is an integral and popular choice for machining many different products. Although well accustomed to machining circular pieces, current CNC lathes are also capable of machining noncircular pieces, usually by halting the turning process and performing milling operations. This produces pieces that can be, for example, circular with flat edges, or have multiple flat sections positioned symmetrically around a common centre. Although these systems are capable of running at high speeds, the work in this thesis is unable to base the speed improvements needed on their design. This is mainly due to the differences in the shape of the work pieces being machined. Although current CNC

lathes are capable of machining non circular parts, they are not completed using turning operations and they are not complex sculptured surfaces.

When looking into the literature for other works that have machined noncircular profiles using a CNC lathe, there are relatively few works to choose from. Frank et al. [8] in 1988 developed a 3-axis turning system that used a grinding wheel to create noncircular work pieces. These parts were variations on common geometric shapes, most often ellipses. This was accomplished by modifying and reprogramming a common cylindrical grinding machine. The motion of the grinding axis was programmed by finding the intersection between the circumference of the grinding wheel and the geometric body. This work shares many aspects in common with the current design of the 3-axis lathe, but it is not capable of creating a complex sculptured surface. Other works have looked at modifying a traditional lathe to cut noncircular shapes. Hwang et al. [11] modified a traditional lathe by attaching a linear motor to the radial x axis of the system. With this modification, they were able to machine noncircular parts, usually of elliptical shape. This system was designed with the purpose of machining automotive pieces, such as cam shafts and piston heads. Although this work had similar ideas to those of the current lathe, they did not attempt and were not able to machine complex sculptured surfaces. Other works [28] have tried developing new software that can be used on a traditional CNC lathe to create noncircular work pieces. They have been successful in creating some noncircular pieces and can even approximate the contour of a square. Again, this work did not attempt to machine any complex sculptured surfaces.

The only other works that seemed to be machining complex sculptured surfaces, and therefore facing similar obstacles to this work, was past research conducted here in the surface machining research group at the University of Waterloo. Manos [15] designed and implemented a single controlled axis lathe that was able to machine relatively simple complex surfaces, such as cylinders with helices on the surface or cylinders with wide, low text on the surface. This was the first design attempt at a lathe capable of machining a sculptured surface and consequently had several limitations. The system used a stepper motor to control the depth of cut of the surface, but the other two axes were moved by hand. This caused surface finish errors when the human operator did not maintain a constant speed, which invariably occurred. The system was designed to be slow moving, hence the human input for rotating the work piece, to allow the stepper motor enough time to adjust the depth of cut of the router. This was necessary because there was no

feedback in the system, which meant that there were no safeguards in place if the stepper motor was to miss its programmed position.

The single controlled axis lathe was later updated by Poon [21]. He took the system and added a second motor which drove both the rotational and translational axes though a coupled pulley and timing belt system. The purpose of this work was to test a new architecture for CNC controllers that was modular in structure and could replace a CAD/CAM package as a single interface. In so doing, they also showed the benefits of machining with more controlled axes, as the test parts produced by the system had a much smoother surface finish that was mostly error free than that of the single controlled axis lathe. This system was also slow moving however, as there was still no encoder feedback for the either of the stepper motors controlling the system. Also, the system was on a smaller scale than the lathe being used in this work, with much smaller motors that limited the speed of the system.

Though the technology of the current 3-axis lathe is based on the previous works in [15, 21], its operation and system mechanics differ greatly. The other works presented, although using somewhat similar research platforms, are using them to accomplish different goals. To improve the overall machining speed of the current 3-axis lathe, the individual components of the system must be considered separately. By considering research in different areas and applying it to the 3-axis lathe system, it should be possible to improve the machining speed of the lathe. The research platform that is the 3-axis lathe and the research goal of improving the machining speed of a complex sculptured surface on this platform is unique enough that there is no certain answer for the best way of achieving this goal.

2.2 Motors

As described in Section 3.2.1, it has been found that the radial axis which controls the depth of cut for the lathe system is the speed limiting axis of the system. The first step towards improving the speed of the machining process was to look into different options for motors to drive the radial axis. Currently a stepper motor is used to control the radial axis of the system. This was a logical choice for the original design of the 3-axis lathes, as stepper motors are commonly used as positioning devices. Stepper motors are well known to provide accurate speed and position tracking with proper control, which is relatively

easy to implement [19]. They can also be a low-cost option in comparison to other motor types [17]. It has been found however, that stepper motors do not perform as well during motions when there are large dynamic requirements [18], such as high accelerations. They are also better suited for low torque applications when compared to DC servomotors [3]. This is believed to be one of the main problems with the 3-axis lathe design, as the radial axis of the system must move approximately 13kg of material a short distance, in a short time, which requires a large acceleration and induces a large torque. If the stepper motor is unable to accurately follow a command that has too high an acceleration, the surface finish will be gouged.

To improve the acceleration capabilities and overall machining speed of the CNC lathe system, a servo motor was selected to replace the stepper motor of the radial axis. Used in many applications, servo motors are able to provide superior torque outputs to stepper motors, with similar if not better positioning performance [3].

2.3 Trajectory Interpolation

An important factor that contributes to the overall speed of the system is the method of interpolating the trajectory profiles of each axis from the toolpath data. Even if a position profile is continuous and smooth, if its derivatives (i.e., velocity and acceleration) are not continuous, then the motor trying to follow the trajectory will have difficulties [25]. This issue arose during the operation of the original 3-axis lathe. The motor controller used to drive the steppers motors of each axis used a linear interpolation technique to create the trajectory profiles for each axis. This means that between each toolpath point a discontinuity existed in the trajectory profile. This discontinuity is responsible for causing instant changes in velocity and acceleration, demanding motions that are impossible for the motors to replicate. Both of these situations caused the stepper motor attempting to follow this profile to miss steps, sometimes causing gouging into the part surface.

A great deal of research has been done in the area of smooth trajectory generation. The obvious benefits are to make the trajectories easier for motors to follow. When the trajectory is smooth and the dynamic demands on motor are within its limits, control signal saturation is avoided. This leads to better positioning control. Having a smooth trajectory also allows the motor to reach higher average speeds. If a motor is being satu-

rated and stalling frequently, its momentum is lost. When the transitions between position locations and the speeds are smooth, the motor can maintain its speed better throughout the motion. An additional benefit is to improve the life expectancy of a servo motor. Avoiding saturation also prevents the motor from overheating during a stall period and potentially damaging the inner windings. As mentioned, smooth trajectories help improve positioning accuracy, this also lends to creating a smoother surface finish, as it helps to avoid discontinuities and jittering effects.

To create a smooth trajectory, the most commonly used technique is the implementation of splines of various forms. Koninckx et al. [13] has used cubic non-uniform rational B-splines (NURBS) to create a trajectory profile that has continuous acceleration. The algorithm developed is intended for real-time calculations of the trajectory, which can be difficult to implement and processor resource intensive depending on the choice of controller. A quintic spline curve, developed in [24], is used to create a smooth trajectory between a given set of data points. This method is said to be an improvement to cubic spline generation because during the chord length parameterization process, cubic splines can cause feedrate fluctuations. This issue is resolved with the use of quintic splines. However, the intended use for this method is high speed precision machining. At the speeds intended in this thesis' work, the magnitude of the feedrate fluctuations discussed in the cubic spline trajectory is not an issue.

Alintas et al. [6] developed a method for feedrate optimization of a quintic spline trajectory. Feedrate optimization in this case is defined as the trajectory profile having continuous jerk motion. This last method is attractive for the 3-axis lathe system. The use of cubic splines for interpolating the toolpath ensures a smooth surface finish. The feedrate optimization technique allows the motors of each axis to push the limits of their dynamics for a specified feedrate. This method of trajectory interpolation was adopted into the operation of the 3-axis lathe system to replace the linear interpolation that was previously being used. The relatively simple implementation and effective results were the main reasons for this choice. A detailed description of this adopted method is contained in section 6.2.

2.4 Smoothing Algorithm

The last method considered for increasing the speed of the 3-axis lathe system is to modify the work piece's model geometry. An algorithm is developed in Section 6.3 that attempts to find the minimum amount of geometry to sacrifice to improve the machining speed of the system. By modifying geometry that causes position tracking errors in a specific way, the errors can be reduced. This allows the system to machine at a faster feedrate without causing gouging in the work piece.

Others have investigated the idea of modifying a model's geometry to achieve specific gains or to meet other criteria. Sun et al. [23] developed an algorithm for modifying segments of a toolpath during the interpolation stage of the machining process. Using the relationship between the geometric and kinematic properties of a curve, segments of the toolpath were identified that would cause the motion planning algorithm to surpass the limits imposed on the system's acceleration and deceleration. The toolpath points of the identified segments were then adjusted by blending linear line segments into the original curve to ensure that the system limits were met.

Erkorkmaz et al. [7] modified geometry to improve the contouring performance around corners for 2D work pieces. Corners of varying angles were smoothed using splines, while ensuring that the features were still within specified dimensional tolerances. Using a virtual CNC machine to simulate the system's contouring performance around the corner, the feedrate was iteratively determined so that the maximum dimensional tolerance was reached, but not exceeded. By altering the geometry of the corner in this fashion, the overall contouring performance was improved in terms of total contouring error. These results were also experimentally verified.

Similarly to these works, this work presents an algorithm that will modify a model's geometry. The goal of this modification, however, is to automatically determine the minimum amount of geometry needed to be sacrificed to machine a satisfactory work piece at a desired feedrate. Contrary to the algorithm presented in [23], it is assumed that the system is already capable of reaching the desired feedrate given the initial model geometry. The problem addressed in this work, is that cutter overshoot in position will cause an unsatisfactory surface finish, in the form of gouging. As shown in [7], the cutter position errors can be reduced by smoothing areas of the model where the overshoot occurs. These areas, which are typically corners, will be identified using a model of the system, inspired by the

virtual CNC used in [7]. The desired feedrate can then be used for machining and achieve the modified surface finish. This algorithm will also produce a preview of the model with modified geometry. This preview will allow a system operator to visually determine if the sacrifices in model geometry are acceptable, or if a slower target feedrate should be used to avoid such heavy sacrifices.

Chapter 3

System Description

The 3-axis CNC lathe that was developed and used as a test platform in this work was created with the purpose of machining sculptured surfaces out of wood. Wood machining was chosen because previous research ([15]) has shown that the cutting force exerted on the tool is approximately 100 N for most types of wood, which is minimal. This small force has little effect on the cutting tool in regards to deflection or restricting cutting speed. This allowed for the design of the system to essentially neglect cutting force constraints that are typically present when designing a system for machining metal. Not having to be concerned with the cutting force exerted on the tool allowed this work to focus on increasing the machining speed of the system and improving toolpath planning. In this work, a sculptured surface is defined as geometric body that can contain large amounts of surface detail that is typically only seen in hand carved designs. Some examples of a sculptured surface would be a surface that contains any type of design, pattern or text. Figure 3.1 illustrates an example sculptured surface. This type of surface is an uncharacteristic product for a manual lathe, as it is difficult to cut. All three axes of a lathe must move together, following nonlinear profiles, to achieve a complex sculptured surface.

In a traditional manual lathe, the work piece rotates at a fixed rpm, frequently in the tens of thousands, while the cutting tool is rigid, only free to move along the length or into the work piece. This configuration essentially makes it impossible for an operator to create a complex sculptured surface. Manual control of the cutter position only allows for symmetric or circular patterns to be cut, such as those typically seen on table legs, which are a common product machined out of wood. Generally, if a non-symmetric surface

Figure 3.1: Sculptured Surface Example

was created on a manual lathe, it would not be repeatable. Traditional CNC lathes, are typically setup to reproduce the same motions as a manual lathe, but in a more precise and repeatable manner. These lathe systems are also unable to produce a complex sculptured surface. They typically machine to quickly to achieve the required level of detail and do not have the necessary degrees of freedom.

The 3-axis CNC lathe designed in this work is able to machine complex sculptured surfaces by slowing down the rotation of the work piece and using a precision controlled motor to control the depth of the cutting tool. A slower rotating part gives the cutting tool the time it needs to move towards and away from the work piece to create the intricate details that exist on a sculptured surface. To ensure that the work piece still receives a clean cut when the work piece is rotating slower, the fixed cutting tool is replaced with a rotating tool. For this purpose a ball nose tool is used with a router to perform the necessary cutting. This setup essentially creates a mill that has the motion capabilities of a lathe.

3.1 System Operation Overview

The following is a summarization of how the 3-axis CNC lathe system operates. Expanded explanations on many of the operations of the lathe are presented in the following sections of this chapter. As with all computer controlled systems, the operation of the 3-axis CNC lathe in this work follows a specific, logical process from the inception of a work piece design, to its creation. Figure 3.2 presents a logic flow diagram that outlines this process. After the idea for a work piece has been conceived, the part must be modelled in software. The system currently works only with the stereo lithography (STL) file format. Presently,

models are created in SolidWorks and then converted into STL format; but any software that can create an STL file is potentially capable of generating a proper model for the system.

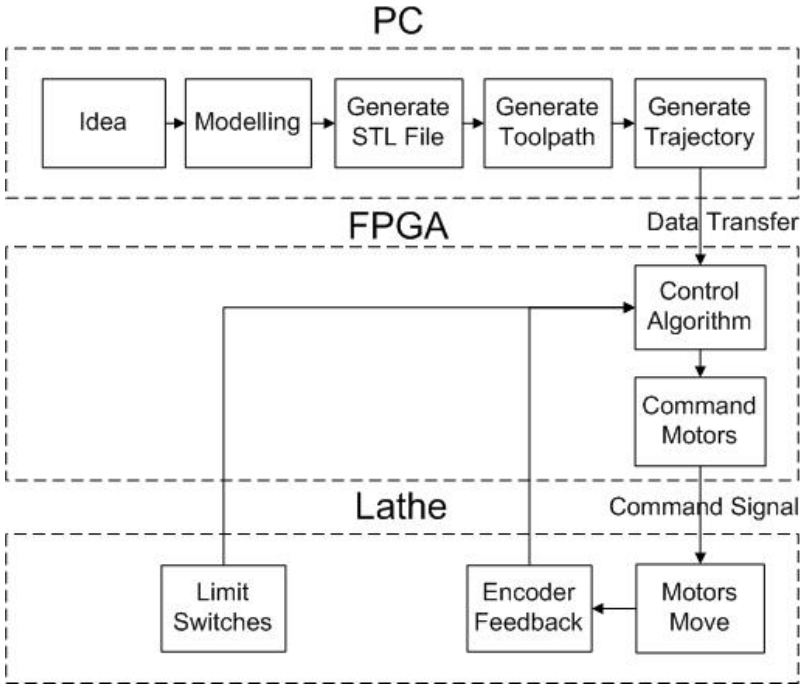


Figure 3.2: Lathe Operation Overiew

A custom toolpath generator, described in section 6.1, is used on the STL file to create a helical toolpath that follows the body of the model, as described in section 3.2.4. The software will generate multiple roughing passes, which remove up to $5mm$ of material at a time, and a single finishing pass, which removes about $1 - 2mm$ of material. The helical shape of the toolpath makes use of the degrees of freedom available to the lathe, as one pass along the helix will allow the cutting tool to reach the entire surface area of the work piece. Next, a custom trajectory interpolator is used to create a position, velocity and acceleration profile for each axis of the lathe system. The algorithm, as described in detail in section 6.2, generates intermediary position points between the toolpath points at an interval that coincides with the sampling time of the control system, which is 1 kHz. The interpolated toolpath points also ensure that the motion of each axis is smooth and does not exceed any of the dynamic thresholds of the system. In other words, it ensures the motion of the tooltip is smooth and therefore the surface finish of the work piece is smooth

as well.

A regular PC is used for modelling the work piece and for the toolpath generation algorithms. The interpolated toolpath information is sent to the control system for the lathe, which runs on an Altera field programmable gate array (FPGA) board. This board, which uses custom designed hardware and software, interfaces with the motor and encoder of each axis to control the lathe. As will be described in section 3.2, two of the axes use stepper motors for operation, while the third uses a servo motor. The axis controlled by the servo motor uses a detailed system model and control algorithm, as described in chapters 4 and 5 respectively, to create an accurate and robust position control system for the lathe. The FPGA board commands the three axes of the lathe using the interpolated toolpath commands as a reference. This will produce the desired work piece model from wood stock when machining.

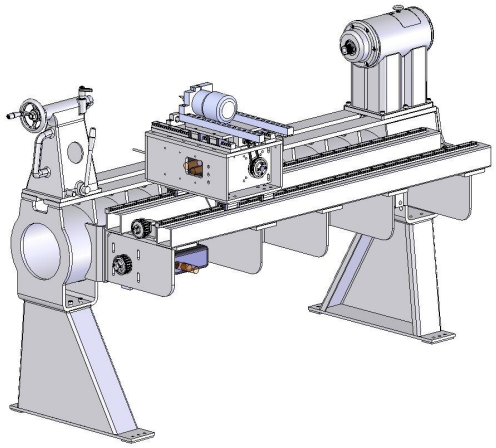


Figure 3.3: 3-Axis CNC Lathe Model

3.2 Mechanical Description

The 3-axis CNC lathe design was created by building off of an existing manual lathe. The decision to build the design up from a manual lathe was made because it would be cheaper than purchasing an existing CNC lathe and it would allow for the flexibility in design that was desired. Most CNC lathes are both expensive and incapable of cutting the intended complex sculptured surfaces. Therefore it was necessary to design our own system, which

was made easier by basing the design on an existing lathe structure. The design was based on the wood turning lathes made by Oneway, who were also contracted to build the base structure of the lathe. As can be seen in figure 3.4, the lathe is made from a steel base that has been both welded and bolted together. The cylinder that makes up the main body of the frame is hollow to help reduce the weight of the system, while its circular profile provides strength. This frame supports the traditional manual lathe components of the system, such as the head and tail stock, between which the work piece is turned. A pair of slide rails that run parallel along the length of the lathe is supported by a number of brackets that are welded to the main cylinder body. The router cutting tool for this system is located on top of a steel box that is attached to the slide rails.

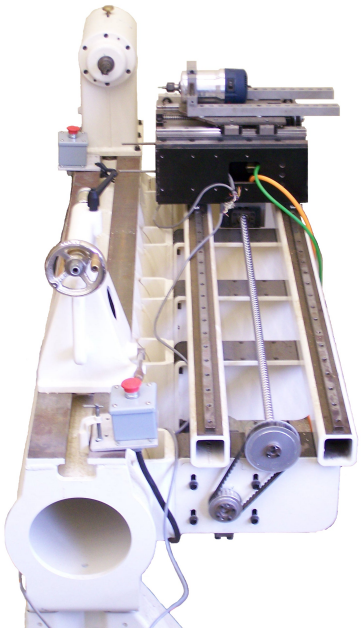


Figure 3.4: 3-Axis CNC Lathe

3.2.1 Axes of Motion

The three axes of the CNC lathe system consist of a translational, rotational and radial axis. These axes, which allow for the same degrees of freedom as that of a manual lathe, are controlled individually to achieve the machining of a sculptured surface. The following section describes each axis in detail.

Translational Axis

The translational axis is depicted in figure 3.5. The main body of this axis is composed of a pair of parallel guide rails that run along the length of the lathe. These rails carry the platform on which the radial axis system is mounted. This setup allows the cutting tool in the router to move up and down the rails, moving along the length of the work piece. The translational motion allows for the work pieces to be cut up to $1100mm$ in length, nearly the entire length of the lathe.

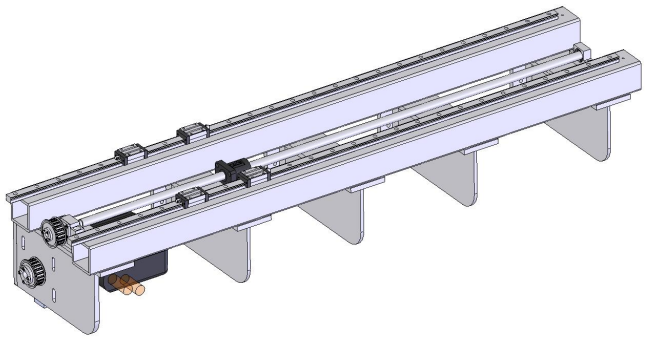


Figure 3.5: Translational Axis Model

The translational axis is driven by a stepper motor and ball screw setup. The stepper motor is an MDrive42, a product of Intelligent Motion Systems Inc., that operates on 120V AC current and comes with a built-in motor controller. This provides an easy interface to the motor, since it only needs digital signals to drive it. This motor is a micro-stepping motor that is capable of turning 200 counts per revolution into 50000 micro-counts per revolution. Selecting a motor with a high resolution for stepping provides greater control of the tooltip motion. For this system a micro-stepping setting of 10000 counts per revolution was chosen, which provides a resolution of $6.176 * 10^{-4}mm$ for the tooltip. This degree of accuracy is appropriate for the aesthetic sculptured surfaces that are being machined on the lathe.

The stepper motor is connected to the ball screw drive by a pair of pulleys and a timing belt. The driving pulley, attached to the motor, has 21 teeth and the driven pulley, attached to the ball screw, has 34 teeth. This provides a gear ratio of 1:0.6176, which is

a reduction in speed for the ball screw and therefore the cutting tool. The speed of the translational axis was reduced based on the intended toolpath design. A helical toolpath implies that to achieve a smooth surface finish with no scallops, the pitch of the helix, or side-step of the cutting tool, must be appropriately small so that the tooltip can remove material from the entire surface of the work piece. A tool used for finishing is typically in the range of 1/8 inch to 1/16 inch in diameter, which corresponds to a helix pitch of approximately half that value. The small movement per one revolution of the work piece suggests that the translational axis is moving much slower compared to the other two axes. Gearing down the axis assists in achieving this motion.

The ball screw on the translational axis that is used to convert the rotational motion of the stepper motor into translational motion of the cutting tool is nearly the same length of the lathe, measuring 1510 mm. The pitch of the ball screw is $10mm$ per revolution. Equation 3.1 combines the pitch of the ball screw, the driven and driver pulley ratio and the number of revolutions of the stepper motor into one expression. This equation represents the conversion from stepper motor revolutions into millimetres, which is a crucial ratio for the control of the translational axis.

$$\underbrace{1\ rev}_{MotorRevs} * \underbrace{\frac{21}{34}}_{PulleyRatio} * \underbrace{\frac{10\ mm}{1\ rev}}_{BallscrewPitch} = \underbrace{6.176\ mm}_{LinearTravel}. \tag{3.1}$$

Rotational Axis

The rotational axis, depicted in figure 3.6, is contained within the head stock of the lathe. This axis of motion is responsible for rotating the work piece. The work piece, which typically has a round cross-section, is centred about the rotational axis during machining. Through the rotation of the work piece, the cutting tool is able to cover the full circumference, or perimeter of the shape.

The rotational axis of the lathe is driven by a stepper motor, the same type of motor as described in the translational axis section above. A stepper motor was selected for the rotational axis because a motor was needed that could follow changing velocity commands. This differs from a regular lathe, where a constant speed motor would suffice. The trajectory interpolator used to construct the motion profiles for all three axes does not turn the rotational axis at a constant speed, but instead slows down and speeds up the axis

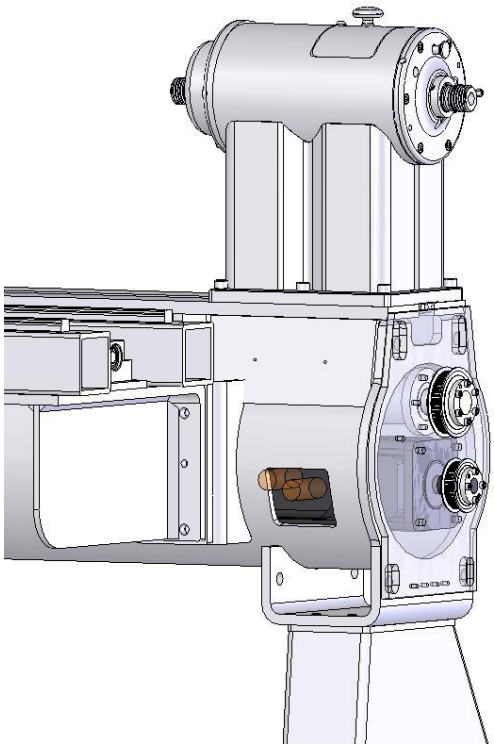


Figure 3.6: Rotational Axis Model

to coincide with geometry complexity. By having a variable speed trajectory for the rotational motion of the work piece, the radial axis is allotted more time to complete its rapid movements. This allows for a smoother trajectory to be followed by all axes, achieving a smooth surface finish and reduces the overall machining time.

The stepper motor drives the rotational axis through a set of four pulleys connected with timing belts, as depicted in figure 3.7. The motor is housed inside the body of the lathe headstock to keep it clean during operation. Due to space restrictions, the motor does not drive the rotational axis shaft directly, but instead requires an additional set of pulleys on the outside of the system to reach an intermediary shaft above the motor. The pulley sizes create a ratio of 1:0.75. The intermediary shaft drives the rotational axis shaft of the lathe through a second set of pulleys and timing belt. These pulleys create a gearing ratio of 1:0.5909. This creates a combined gear ratio of 1:0.4432 for motor rotation to axis rotation amount. As with the translational axis, the micro-stepping resolution of the stepper motor for the rotational axis is set to 10000 counts per revolution. To convert the rotations of the motor into degrees of rotation of the rotational axis, the motor information must be coupled with the gearing ratio information of the timing pulleys. This relationship is expressed in equation 3.2 below.

$$\underbrace{1\ rev}_{MotorRevs} * \underbrace{\frac{26}{44} * \frac{36}{48}}_{PulleyRatio} * \frac{360\ deg}{1\ rev} = \underbrace{159.545\ deg}_{RotationalTravel} . \tag{3.2}$$

Radial Axis

The radial axis of the lathe system is depicted in figure 3.8. The structure for this axis of motion is built atop the steel box shown in the figure and is attached to the guide rails of the translational axis. As the translational axis moves, the entire radial axis is moved with it. The radial axis consists of a steel plate that sits on two guide rails that run the length of the box. A ball screw is used to drive the plate forwards and backwards. The router, used for cutting the wood surface, is attached using a custom clamping system to the steel plate. By moving the steel plate back and forth along the guide rails, the router and therefore the cutting tool, is moved, controlling the radial depth of cut into the work piece.

To drive the radial axis, a servo motor is attached to the ball screw system. The motor is positioned underneath the radial axis inside the steel box shown in figure 3.8 and drives

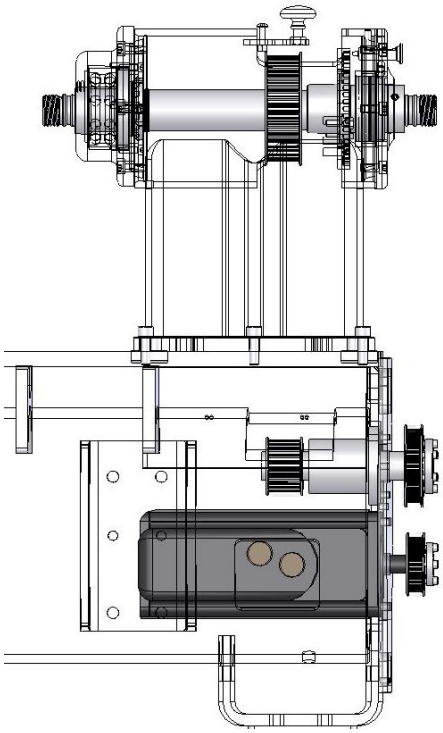


Figure 3.7: Rotational Axis Pulley System

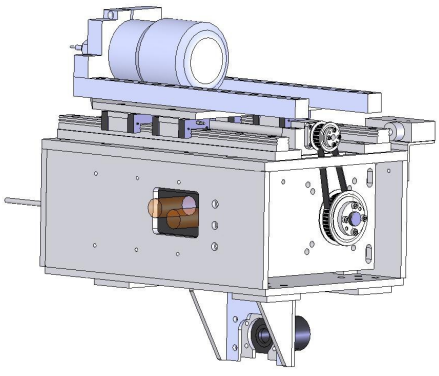


Figure 3.8: Radial Axis Model

the ball screw using a pulley and timing belt system. Originally in previous versions of the lathe, a stepper motor was used to control the radial axis. However, the radial axis has been determined to be the speed limiting axis of the system. This means that the radial axis experiences higher accelerations and torques than the other two axes during machining. When speeding up the combined feedrate of the three axes during machining, it is the radial axis that prevents the system from going faster. Due to the complex sculptured surfaces of most work pieces, the trajectory of the radial axis contains a large amount of in and out oscillatory motion. When using a stepper motor for controlling the radial axis, the nature of the motor prevents it from achieving the high accelerations in the short time required to successfully follow the trajectories without slipping. The feedrate of the machining process must be slowed to allow the motor to follow the trajectories correctly. To improve the response times of the radial axis, a servo motor was selected to replace the stepper motor. A servo motor is able to achieve higher accelerations in a shorter amount of time than the stepper motor. This will increase the speed limit that exists on the radial axis and allow the feedrate of the machining process to be increased. A model and control algorithm is required to operate a servo motor and these will be presented in Chapters 4 and 5 respectively.

The servo motor that was selected was a Baldor BSM80C-275AF. This motor operates on 220V AC and is capable of running at 4000 rpm and outputting 2.4 Nm of torque. It requires an analog input of ±10 volts and has a built-in encoder with an output resolution of 10000 counts per revolution. The pulley and timing belt system linking the motor to the ball screw create a ratio of 36:28. This ratio was initially intended to be 1:1, but size constraints on the pulley at the ball screw end of the connection prevented the pulleys from being the same size. The pitch of the ball screw is 5 millimetres per revolution. By combining this information, the conversion of one revolution of the servo motor into linear motion of the tool tip can be calculated. This expression is summarized in equation 3.3.

$$\underbrace{1\ rev}_{MotorRevs} * \underbrace{\frac{36}{28}}_{PulleyRatio} * \underbrace{\frac{5\ mm}{1\ rev}}_{BallscrewPitch} = \underbrace{6.426\ mm}_{RotationalTravel} . \tag{3.3}$$

3.2.2 Coordinate System

Each axis of motion described in the above section is assigned a positive and negative direction, along with an axis label. Figure 3.9 below depicts the coordinate system for the

3-axis CNC lathe system. The point of origin for the coordinate system originates at the head stock of the lathe, at the centre of the rotational axis where the work piece is fixed prior to machining. The translational axis is denoted as the z-axis. The positive direction is out from the origin towards the tail stock of the lathe, which is parallel to the ball screw of the translational axis. The rotational axis is labeled the c-axis. This axis rotates in the positive direction clockwise around the z-axis. The radial axis is classified as the x-axis. The positive direction of this axis is moving from the physical cutter tool tip towards the centre of rotation of the work piece.

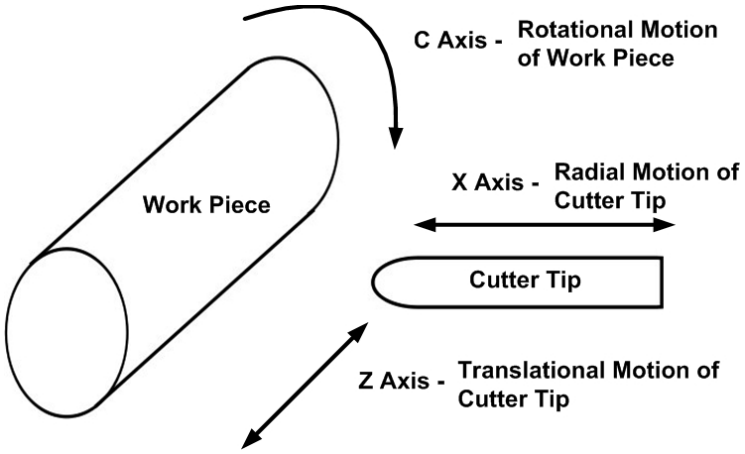


Figure 3.9: Lathe Coordinate System Diagram

3.2.3 Cutting Tool

In a manual lathe, the cutting tool is typically a fixed blade that cuts when it is pushed into the rotating work piece. The 3-axis CNC lathe however, uses a router and ball nose tool to cut the work piece instead of a fixed tool. This modification was made because manual lathes typically spin at fast rpms, often in the thousands, which allows a fixed edge to cut the work piece. The 3-axis CNC lathe on the other hand rotates slowly by comparison, typically around 10 rpm prior to this work's modifications. When rotating slowly, an immobile tool would be unable to cut the surface of the work piece without tearing. A router, which rotates the tool, solves this problem. The slow speed of the rotational axis is necessary because it must move slow enough to allow the radial x-axis time to complete its fast in and out motions. As stated in section 3.2.1 above, the radial

axis is the limiting axis of the system. The speeds of all other axes are dictated by the speed of the radial axis.

3.2.4 Toolpath

Although the CNC lathe is able to move in a variety of directions, as described in Section 3.2.1, the toolpath that is machined on the system is a helical toolpath, similar to the path followed by a manual lathe. This type of path is used because a custom toolpath generator has been developed that can convert a SolidWorks model into a helical toolpath of points [15]. This toolpath generator is necessary, as all of the software for the lathe is custom made. To use standard CAD/CAM packages would require extra work to interface with the CNC lathe system, as well as add to the cost of the system.

As seen in figure 3.10, the helical toolpath starts at the end of the work piece that is closest to the head stock of the lathe. The helix then spirals around the work piece towards the opposite end of the lathe. To accomplish this movement, the rotational axis will spin the work piece and the translational axis will slowly move the steel box structure supporting the cutting tool along the length of the work piece. This motion will trace out a spiral of uniform radius. To achieve a sculptured surface, the radial axis will move the cutting tool in and out to alter the depth of cut of the tool as the helix is traversed. The toolpath generator described in section 6.1 produces multiple roughing passes and a finishing pass. Typically a roughing pass will use a ball nose tool with a radius of $3.175mm$ to remove material, approximately $5mm$ depth at a time. If a larger tool is used, machining out details in the sculptured surface may be impossible. The side step of the helix for a roughing pass is calculated to be equal to the radius of the tool, so that material from the entire surface of the work piece is removed. The finishing passes use a smaller radius ball nose tool, typically $1.5875mm$. This allows the tool to machine a greater amount of detail in the sculptured surface than during the roughing passes. The side step of the helical toolpath during a finishing pass is calculated to be smaller than the radius of the tool. A ball nose tool will typically leave behind a ridge of material, called a 'scallop', because the tool is round and will only contact the surface of the work piece at a single point. By making the side step of the helix less than the radius of the tool being used, these scallops are machined off, leaving a relatively smooth surface.

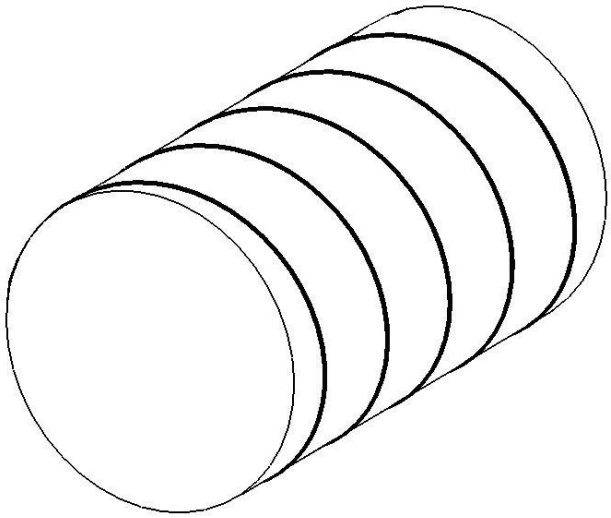


Figure 3.10: Helical Toolpath Example

3.3 Electrical Description

The electrical system of the 3-axis CNC lathe has been custom designed and integrated with the mechanical system. This electrical system links a regular desktop PC, to a motor control system, to the motors controlling each of the three axes. A return loop connects the encoders attached to each motor and brings the data back to the motor control system. The following describes the components that make up the electrical system of the 3-axis CNC lathe.

3.3.1 Computer Interface

Currently, the interface with the CNC lathe uses a standard desktop personal computer (PC). This computer is used for calculating and sending the necessary data required for machining the work piece to the motor control board. Currently the PC runs a Microsoft Visual Studios program to take an STL file and generate an appropriate toolpath for the work piece, as described in section 6.1. A script in Matlab is run next to generate a trajectory for each axis that follows the set toolpath at a specified feedrate. This trajectory generating algorithm is described in detail in section 6.2. This trajectory data is then sent to the motor control system using a serial port for communication.

3.3.2 Motor Control System

The motor control system for the 3-axis CNC lathe actually consists of several components that are used together to convert the mathematical data describing the work piece provided by the PC, into electrical signals that drive the lathe's motors to machine a work piece. Figure 3.11 contains an overview of the motor control system. The main component of the motor control system is the field programmable gate array (FPGA) board Stratix II. This board is a product of the company Altera and is used as a prototyping board for many different hardware applications. It supports both custom hardware and software design, allowing a user to fully customize and test their design before manufacturing the final custom PCB that has been tailored to their specific application.

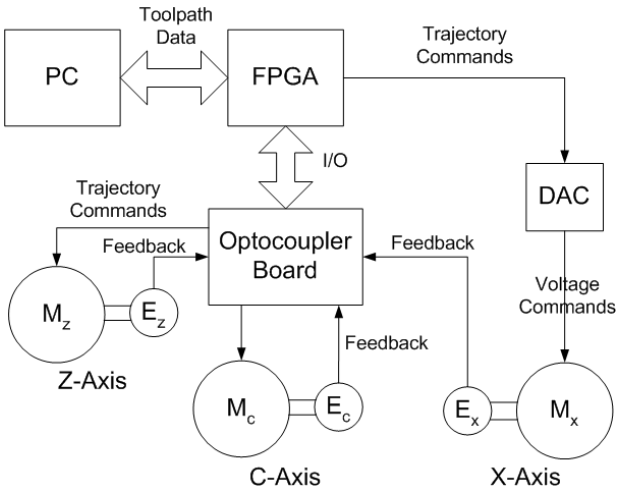


Figure 3.11: Motor Control System Overview

In the previous design iteration of the CNC lathe, a pre-made motor controller card, called a PMAC, was used to operate the lathe. However, this card had several disadvantages that led to the switch to an FPGA board. The PMAC only accepted data points with timing constraints and performed its own interpolation. Linear interpolation was used, which caused problems for the motors at higher speeds due to the velocities and accelerations not being continuous. As described in section 2.3, a new interpolation method had been selected that creates smooth trajectories that can be more easily followed by the motors and consequently improve the machining speed of the system. This interpolation method could not be implemented on the PMAC. Also, a pre-made motor controller has a

fixed amount of motor output channels. The PMAC had enough channels to control three motors. However, as mentioned in Future Work in chapter 8, the next progression in this research is to expand the 3-axis CNC lathe into a system that has four or even five axes. Such a system would require a new pre-made motor controller capable of interfacing with the necessary number of motors. An FPGA board however, has a customizable number of inputs and outputs, allowing the control capabilities of the board to be expanded easily through software settings. Finally, the PMAC is also expensive. Such a controller for three or more motors costs thousands of dollars. If the 3-axis CNC lathe system is commercialized, purchasing a pre-made controller board will add to the manufacturing cost of the system. By prototyping with an FPGA board, when the design is complete, a PCB can be manufactured at a much lower cost than that of purchasing an off-the-shelf product.

The FPGA board comes with a small processor that can be customized to work with different hardware peripherals that are included in the prototyping board. The main hardware components that are currently in use are the serial port for communication with the PC, and a number of digital input/output (I/O) ports that are used to communicate with the motors, encoders and limit switches attached to the lathe. This processor runs custom C software and can interact with the configured hardware peripherals to control the CNC lathe system. Currently, the FPGA board will wait for the trajectory data to be loaded from the PC using the serial port. The data for the entire work piece is loaded into the board all at once, limiting the size of the work piece by the onboard memory capacity of the FPGA processor. As mentioned in section 8.2, this method will be modified to transfer sections of the toolpath during machining, which will increase the potential length of the work piece to the physical limits of the lathe system. Once the command for machining to begin has been received, the FPGA board software will step through the trajectory data incrementally, commanding the lathe using I/O ports and monitoring the feedback from the motor encoders and limit switches. The control strategy developed in section 5 is implemented in the software and applied to the trajectories during the machining process.

To connect the FPGA board with the motors, an additional PCB board is used to isolate the FPGA. This board contains optocouplers that separate the digital signals between the FPGA and motors by using light to relay the signal. This precaution was taken to protect the FPGA from any current spikes that may come from the electric motors. The FPGA is unable to sink a large amount of current and could easily be damaged if the motor it is controlling produced a surge.

The last component of the motor control system is a PCB with both an analogue-to-digital converter (ADC) and a digital-to-analogue converter (DAC). This board is required to control the servo motor that drives the radial axis of the lathe. The amplifier that provides the current to the servo motor requires a voltage input signal of ± 10 volts. The FPGA output ports are low voltage, or digital outputs, and cannot source the necessary voltage. The DAC is used to convert the digital signal into the appropriate voltage value. This conversion was done using 10 digital channels, giving the output voltage a minimum step size of 19 mV. The ADC was intended to be used to convert the output from an accelerometer into a corresponding digital signal. However, this sensor was never installed during the course of this research and is included in the future work in section 8.2.

Chapter 4

Servo Motor Modelling

Having an accurate model describing a system provides many advantages and is, for all intents and purposes, required for the development of more sophisticated control schemes. In the previous version of the lathe system, the control method was tuned experimentally. This was done largely by trial and error, and could be potentially time consuming without any confirmation whether the final configuration was optimal. Using a model allows the controller to be designed mathematically based on the parameters of the model, before the controller is implemented experimentally. The theory behind most control methods is heavily based on the parameters of the model; so the more accurate the model, the better the performance from the controller. The biggest advantage of an accurate model however, is the ability to simulate the output of a system. Once a model has been validated to show that the simulated output matches the experimental output of the system to within a desired tolerance, the development time of a controller is greatly reduced. A model allows the controller to be tuned quickly and efficiently before being validated on the physical system. Although simulation is important, the model, and therefore the simulation of the system output, will invariably differ to some degree from the actual system. This makes experimental validation necessary.

This chapter discusses the development of the system model used to describe the x-axis of the 3-axis CNC lathe system, which is depicted in figure 3.9. Section 4.1 rationalizes the selection of a linear 1st-order model for the x-axis. The dynamic equations used to describe the x-axis as a contained input/output system and the simplifications made to that model are in section 4.2. The parameters of the model are identified and a friction

model is developed for the axis in sections 4.3 and 4.4 respectively. Finally, the axis model is validated in section 4.5 and then summarized in 4.6.

4.1 Model Selection

The selection of the model structure is an important step. There are many common model types being used today for system identification, with some being more suitable than others depending on the goal of the model. All models fall into one of two main categories, either parametric or non-parametric [26]. The non-parametric methods are usually used when no prior information about the system is known. Experimental data, gathered from the system and manipulated in specific ways, can be used to create a model. The resulting model is a parametric model, but the structure is entirely dependent on the experimental data. For the parametric methods, some information about the system is either known or assumed. The structure of the model (i.e., the model type and order) is selected prior to analysing the experimental data. The parameters for the selected model structure are then estimated by analysing the experimental data.

As discussed in section 1.7, the model in this work is required for the x-axis of the system, which consists of a table and ball screw mechanism, driven by a servo motor. The servo motor has been the subject of modelling and control for many years. As a result, there are numerous sources [12] documenting the successful modelling of different systems, using many of the now common model types and structures. These previous works present a good knowledge base for the decision making required in the identification of this work's system. This knowledge base allowed for a parametric model to be used.

The model selected for the lathe system's x-axis was a linear first-order transfer function model. Although servo motors are non-linear mechanical systems, they are often approximated as linear systems [12, 10, 5]. Different nonlinear models do exist, but they can be complicated and are frequently linearized about an operating point for simplification [16]. The control laws that have been developed for nonlinear systems are also complex. As it is the goal of this work to develop a suitable controller and model for the servo motor axis, the time requirement for an accurate nonlinear model and controller was not justifiable. Therefore the simpler yet sufficiently accurate linear model was selected. The transfer function structure that was selected for the model is probably the most common model

structure in use. It is easy to switch to other useful forms, such as state space or discrete time models, which are useful for controller design and computer implementation respectively. The transfer function structure has other advantages. Several of the most common identification techniques, such as least squares, work well with the transfer function form. This form can also be derived directly from the dynamic equations that describe a system, or used to describe specific aspects of the dynamics. This last property is the main reason the transfer function structure was selected, as the knowledge base of servo motors and ball screw drives available will be best utilized with the transfer function structure. The model structure was selected to be first-order because, as will be shown, the basic dynamics of the mechanical system of the ball screw can be described by using a first-order model. As shown in [5], a first order model of a servo motor and ball screw drive is sufficient to achieve reasonable position tracking, which is a goal of this work.

4.2 Model Development

The modelling techniques presented in the remainder of this chapter are derived from [4, 5, 14, 26]. Figure 4.1 pictures the dynamic model that is used to describe the servo motor and ball screw system. Table 4.1 lists the names and units of the corresponding variables in the diagram. This dynamic model was adapted from the model provided in [5], which was also for a servo motor and ball screw system, and therefore required little modification.

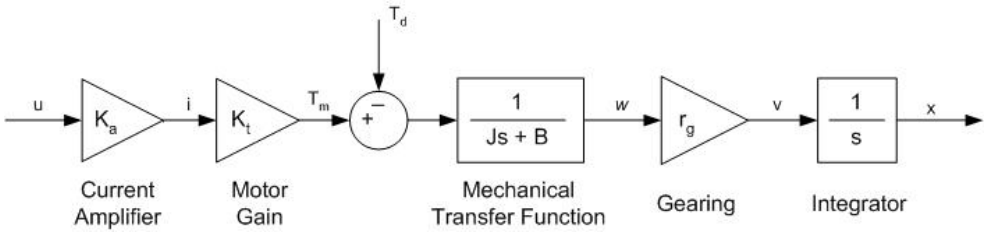


Figure 4.1: Dynamic System Model

In accordance with the model type selected above, the amplifier and servo motor gains are treated as linear constants, represented by the variables K_a and K_t respectively. The

Variable	Description
u	Control Signal $[V]$
K_a	Amplifier Gain Constant $[A/V]$
i	Motor Current $[A]$
K_t	Motor Gain Constant$[Nm/A]$
T_m	Motor Torque $[Nm]$
T_d	Disturbance Torque $[Nm]$
J	Rotational Inertia $[\frac{Nm}{Rad/sec^2}]$
B	Viscous Damping $[\frac{Nm}{Rad/sec}]$
ω	Rotational Velocity $[Rad/sec]$
r_g	Gear/Ball Screw Ratio $[\frac{m}{Rad}]$, $[\frac{mm}{Rad}]$
v	Linear Velocity $[\frac{mm}{sec}]$
x	Linear Position $[mm]$

Table 4.1: Dynamic System Model Variables

output from the servo motor is a motor torque (T_m), which is assumed to be affected by an external disturbance torque (T_d). The disturbance torque could be caused by factors such as the systems inertia, noise, friction and gear backlash. This resultant torque is then transferred to the ball screw, which is represented as a first-order dynamic system. The assumption made is that the timing belt connecting the motor and ball screw form a rigid body. This allows the motor to directly affect the rotating mass of the ball screw and the translating load, which can be represented by a first-order transfer function [4, 5]. The rotational inertia (J) and viscous damping (B) parameters are used to described the first-order system in the transfer function block pictured above. The rotational velocity (ω) is the ball screw output and is converted into linear motion by using the linear constant (r_g), which represents the gear ratio between the motor and ball screw shaft, and the ball screw pitch which moves the attached table. The linear velocity is converted into a linear position (x) by an encoder that captures the output of the system.

The model is then simplified into a control signal equivalent form. This form, shown in figure 4.2 below, reduces the complexity of the previous model, making controller development simpler and leaving fewer parameters to model. Simple block diagram manipulations

were used to reach this new form. Table 4.2 contains the new variable descriptions and simplifications.

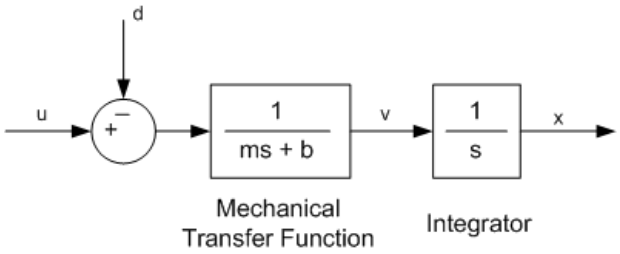


Figure 4.2: Simplified Dynamic System Model

Variable	**Simplificaiton**	**Description**
u	-	Control Signal $[V]$
d	$\frac{T_d}{K_a \cdot K_t}$	Control Signal Equivalent Disturbance $[V]$
m	$\frac{J}{K_a \cdot K_t \cdot r_g}$	Control Signal Equivalent Inertia $[\frac{V}{mm/sec^2}]$
b	$\frac{B}{K_a \cdot K_t \cdot r_g}$	Control Signal Equivalent Viscous Damping $[\frac{V}{mm/sec}]$
v	-	Linear Velocity $[\frac{mm}{sec}]$
x	-	Linear Position $[mm]$

Table 4.2: Simplified Dynamic System Model Variables

The last step to prepare the model for identification is to transform the model into a discrete-time equivalent form. Although the continuous form can be used for controller development and simulation, it is best to use a discrete model when identifying the parameters, as the experimental data being used is discrete itself. To accomplish this, the equation represented by the block diagram in 4.2, before integration occurs, was rewritten as,

$$v(s) = \frac{\frac{1}{m}}{s + \frac{b}{m}} \cdot (u(s) - d(s)), \tag{4.1}$$

where $\frac{1}{m}$ can be represented by the gain K_v and $\frac{b}{m}$ can be represented by the pole P_v. A z-transform can applied to equation 4.1, using a zero-order hold of the form $\frac{1-e^{-sT_s}}{s}$. This

transformation produces the equation

$$v(z) = \frac{K_v}{P_v} \cdot \frac{1 - e^{-P_v T_s}}{z - e^{-P_v T_s}} \cdot (u(z) - d(z)). \tag{4.2}$$

For further simplification, a discrete-time gain and pole variable can be defined as,

$$K_{vd} = \frac{K_v}{P_v} \cdot (1 - e^{-P_v T_s})$$

and

$$P_{vd} = e^{-P_v T_s}$$

respectively. The allows equation 4.2 to be rewritten as,

$$v(z) = \frac{K_{vd}}{z - P_{vd}}(u(z) - d(z)). \tag{4.3}$$

The final simplified discrete-time model of the x axis, shown in (4.3) can be used for identification of the continuous system parameters with discrete experimental data. The continuous-time model parameters m and b can be equated to K_{vd} and P_{vd} directly through back substitution.

4.3 System Identification

One of the most common methods used for system identification is least squares. In general, least squares takes experimental data and creates a line of 'best fit' through the data, based on a specified model structure and order. The method allows for many different model structures and orders, but always arrives at the same general solution form [26]. Given the discrete transfer function developed in (4.3) above, the difference equation model form of least squares can be used and easily implemented. This form, as seen in [14], takes the discrete experimental data of the system, and uses it to determine the value of the unknown model parameters. The model in equation 4.3 must be rearranged into a difference equation form to use the intended least squares variation. To achieve equation 4.4, the 'z' terms in equation 4.3 are treated as shift operators, which offset a set of discrete data by one point into the future when multiplied by the shift operator. The 'k' index represents as a single discrete data point in a data set.

$$v(k+1) = P_{vd} * v(k) + K_{vd} * u(k) - K_{vd} * d(k) \tag{4.4}$$

The disturbance term d in equation 4.4 is assumed to be dominantly comprised of the friction present in the system. Figure 4.3 shows the Stribeck friction curve [5], which describes the relationship of static, viscous and coulomb friction to velocity. The disturbance in the model is approximated by a constant plus/minus value, representing the coulomb friction portion of the Stribeck curve. Although the system parameters could be identified while ignoring the disturbance input, including an approximation for the disturbance will improve the estimates of the system's inertia and viscous damping parameters. Using coulomb friction as an approximation for the disturbance input is appropriate because coulomb friction accounts for the largest part of the Stribeck friction curve. In this way, a simple estimate for friction can be used to improve the parameter estimates of the system model. A more rigorous friction model will be introduced afterwards in section 4.4 to improve the performance of the system model. Equations 4.5 to 4.8 below show the

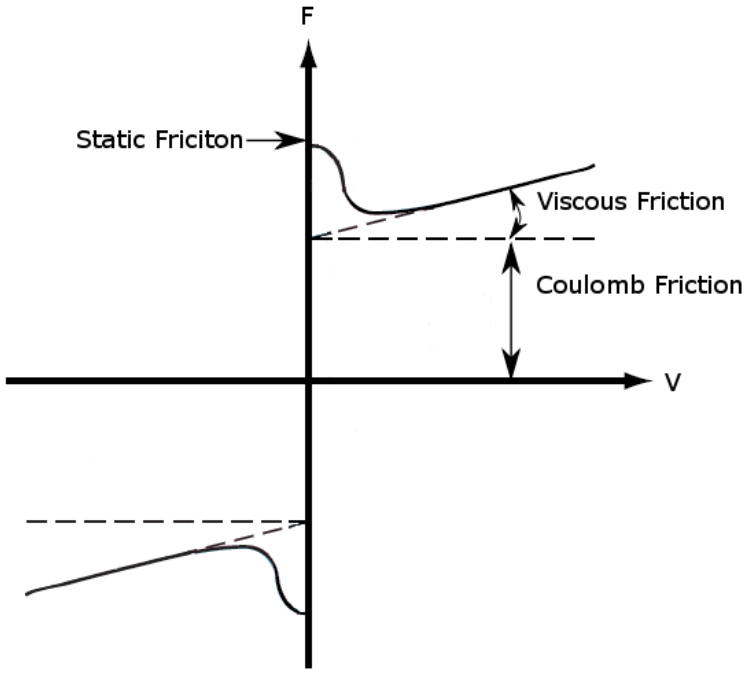


Figure 4.3: Stribeck Friction Curve

switching function used for determining the constant value of coulomb friction, depending on the direction and magnitude of the system's velocity. As indicated in the equations, the disturbance term will only have one value at a time, as the friction switches between the lower value (d_c^+), zero and the upper value (d_c^+). When collecting the experimental data,

the switching value of the $PV(v)$ and $NV(v)$ terms will be the known data.

$$\sigma(v) = \begin{cases} +1, & \text{if } v > v_{min} \\ -1, & \text{if } v < -v_{min} \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

$$PV(v) = \frac{1}{2}\sigma(v)(1 - \sigma(v)) \tag{4.6}$$

$$NV(v) = -\frac{1}{2}\sigma(v)(1 - \sigma(v)) \tag{4.7}$$

$$d = PV(v(k)) \cdot dc^{+} + NV(v(k)) \cdot dc^{-} \tag{4.8}$$

Equation 4.9 shows the modified difference equation with the coulomb friction estimate used as the disturbance input. The $K_{vd} \cdot dc^{+}$ and $K_{vd} \cdot dc^{-}$ terms are the additional unknown parameters that must be identified.

$$v(k+1) = P_{vd} \cdot v(k) + K_{vd} \cdot u(k) - PV(v(k)) \cdot K_{vd} \cdot dc^{+} - NV(v(k)) \cdot K_{vd} \cdot dc^{-} \tag{4.9}$$

The difference equation can now be expressed in matrix form, defined by the least squares method, to represent all of the input and output data collected during an experimental test. A general form of this matrix is shown in equation 4.10 below. The matrix form is separated into inputs and outputs, and known and unknown parameters. By solving this system of equations in matrix form, the unknown parameter vector θ, which contains the unknown model parameters, can be solved. The solution to this matrix form can be found using equation 4.11, known as the left pseudo-inverse equation. In this way, the least squares method uses the trends of all of the input and output data of the system to estimate the parameters defined by the dynamic model of the system.

$$\underbrace{\begin{bmatrix} v(2) \\ v(3) \\ \vdots \\ v(N) \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} v(2) & u(1) & -PV(v(1)) & -NV(v(1)) \\ v(3) & u(2) & -PV(v(2)) & -NV(v(2)) \\ \vdots & \vdots & \vdots & \vdots \\ v(N) & u(N-1) & -PV(v(N-1)) & -NV(v(N-1)) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} P_{vd} \\ K_{vd} \\ K_{vd} \cdot dc^{+} \\ K_{vd} \cdot dc^{-} \end{bmatrix}}_{\theta} \tag{4.10}$$

$$\theta = (\Phi^T \Phi)^{-1} \Phi^T Y \tag{4.11}$$

Figure 4.4 shows the input signal that was used for the purpose of system identification. This signal, an oscillating square-wave, has some unique characteristics that make

it appropriate for system identification. The units of the input signal are voltage, which is supplied to the motor amplifier. This input demands a constant torque output from the motor. Constant torque, which is proportional to constant acceleration, will generate a continuously increasing velocity. By oscillating the input signal between voltages of different magnitude and polarity at a fast interval of 0.1 seconds, the system will never reach a constantly increasing velocity. This is important, as it is observing how the system responds to changes (i.e. the system dynamics) which allows the identification to be successful. If the system output did reach a constantly increasing velocity, that portion of the data would be unusable for the least squares calculation. The amplitude of the square wave was selected to ensure that the effects of static and coulomb friction were observed. If the voltage input is too high, the system will generate an inertial force large enough that the effects of friction are hidden. Figure 4.5 shows the experimental velocity output signal produced by the described input. As can be seen, the velocity in the figure never reaches a steady state of increase, as desired. The data can now be used to estimate the model parameters with the least squares method.

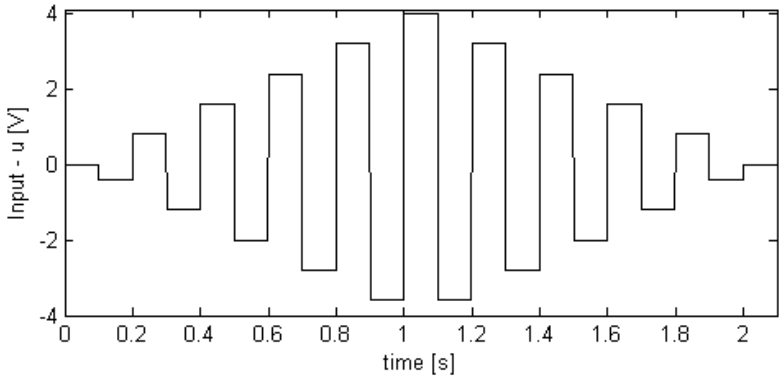


Figure 4.4: Square Wave Input for System Identification

The data from the experimental trial is placed into the matrix form presented in equation 4.10. The solution is solved using equation 4.11 in Matlab. Now with the parameter values of P_{vd} and K_{vd} identified, simple back substitution can be used to solve for the continuous time parameters of inertia (m) and viscous damping (b). Table 4.3 below shows the identified values of the inertia, viscous damping and coulomb friction.

Using the above identified parameters with the model presented in figure 4.6, the system output is simulated to check the validity of the identified parameters. The only change to

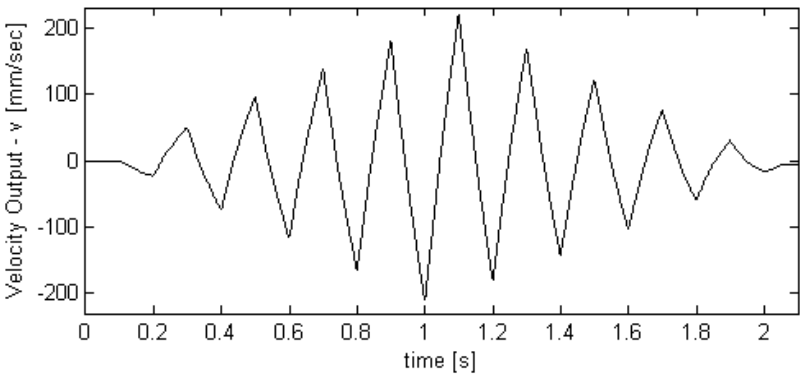


Figure 4.5: Velocity Output of System Identification Test

Parameter	**Value**
Inertia (m)	0.009118
Viscous Damping (b)	0.002577
Pos. Coulomb Friction (dc^+)	0.1273
Neg. Coulomb Friction (dc^-)	-0.09789

Table 4.3: Identified Model Parameters with Coulomb Friction

this new system model from the model in figure 4.2 is the addition of the coulomb friction estimate as the disturbance input. Figure 4.7 shows the results. As can be observed, the simulated output matches closely to the experimental. Figure 4.8 highlights the differences in the outputs by showing a zoomed-in portion of the data at lower velocities. Although the simulation approximates the experimental data well, there is some bias at low velocities. This is attributed to the simple coulomb friction model that is being used. The model does not account for the static friction present at lower velocities. To improve the model further, a new friction model will be developed that more closely matches the Stribeck curve.

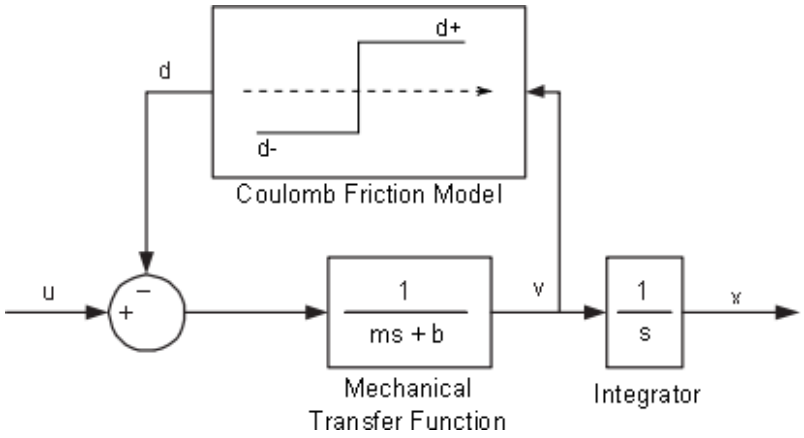


Figure 4.6: System Block Diagram with Coulomb Friction Model

4.4 Nonlinear Friction Modelling

The new friction model that will be developed to improve the system model's accuracy is a nonlinear friction model. This model, as seen in [5], produces a better representation of the Stribeck curve by using equation 4.12. Table 4.4 describes the variables used in (4.12).

$$d = \begin{cases} d_c^+ + (d_s^+ - d_c^+)e^{-|v|/\Omega} \\ d_c^- + (d_s^- - d_c^-)e^{-|v|/\Omega} \end{cases} \tag{4.12}$$

This equation uses estimated values of both static and coulomb friction, and takes into account the viscous damping parameter from the model. To identify the parameters of

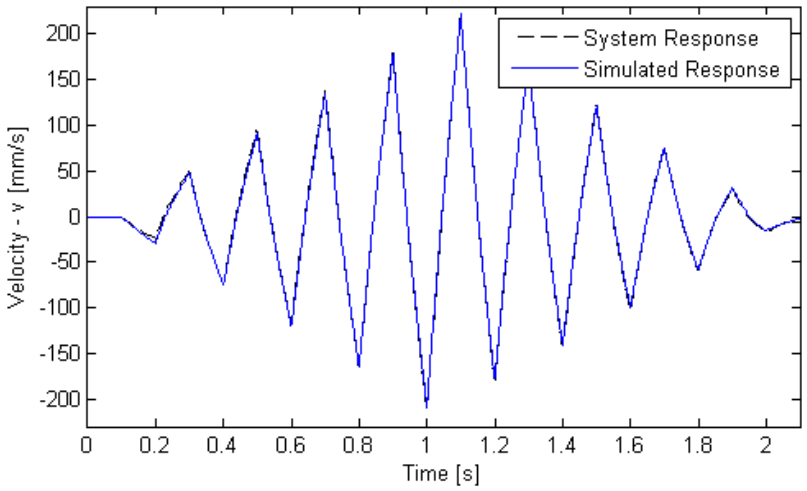


Figure 4.7: Model Validation - Simulated vs. Experimental System Response

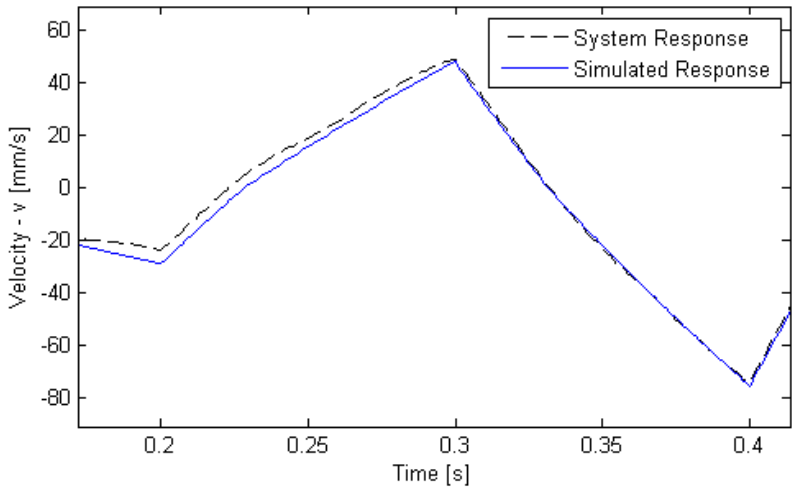


Figure 4.8: Model Validation - Simulated vs. Experimental System Response, Zoom

Variable	Description
d_c^+	Pos. Coulomb Friction
d_s^+	Pos. Static Friction
d_c^-	Neg. Coulomb Friction
d_s^-	Neg. Static Friction
Ω	Velocity Constant

Table 4.4: Nonlinear Friction Model Parameter Description

this friction model, a Kalman filter is used to estimate the disturbance observed during a constant velocity test. The velocities must be constant so that any fluctuations away from the desired velocity can be identified as a disturbance and not a dynamic response. By estimating the disturbance at a number of different velocities, a 'disturbance vs. velocity' plot can be generated. As stated earlier, by assuming the disturbance observed in the system is primarily attributed to friction, this plot will represent a friction versus velocity profile. The unknown parameters can then be estimated based on this plot. By identifying these friction parameters accurately, equation 4.12 should approximate the experimental friction profile.

4.4.1 Kalman Filter

The Kalman filter developed for this identification process is based on the work in [5]. It uses a discrete time state space form, with the states for the x-axis selected as position and velocity, which are logical system outputs. The transfer function form of the system model is re-written into continuous time state space form as

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & \frac{-b}{m} \end{bmatrix}}_{A_c} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{B_c} u(t) \\ [y] &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C_c} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \underbrace{[0]}_{D_c} u(t) \end{aligned} \quad . \tag{4.13}$$

The above model can be transformed into the discrete time domain by using the transformation [9],

$$A = e^{A_c T_s}. \tag{4.14}$$

In this equation, Ts is the sampling time of the discrete system and A_c represents the continuous state space matrix. A similar transformation to equation 4.14 can be applied to each matrix in the continuous state space model to complete the transformation into a discrete time model. This yields a discrete time model of the form,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}. \tag{4.15}$$

The Kalman filter creates estimates of the states based on the provided constant velocity input and output data. To obtain an estimate of the disturbance observed during movement of the x axis, a dynamic model for the disturbance must be added to the state space model shown in equation 4.15 above. The disturbance is given the form

$$d(k+1) = d(k) + \omega_d(k). \tag{4.16}$$

This model assumes that the disturbance is a sequence of piece-wise constant signals. The term ω_d is a perturbation to the disturbance process. This perturbation is assumed to be a normal distribution with a zero mean and a variance $R\omega_d$. The variance for the disturbance model is a tuning parameter for the activeness of the Kalman filter. By increasing $R\omega_d$, the performance of the Kalman filter should improve, as the Kalman filter will be expecting a greater amount of disturbance. The disturbance estimate will become 'jittery' when the value of $R\omega_d$ is too high, however, which is an indication of a maximum value. The value 0.008 $[mm^2]$ was determined experimentally to be suitable for this system. Figure 4.9 shows the disturbance predicted by the Kalman filter with the chosen variance.

The output state equation must also be modified to account for the disturbance parameter added to the system's state vector. The disturbance does not directly affect the output however, so the output state equation will remain similar to equation 4.13, although the matrix dimensions must be modified to account for the disturbance parameter in the state vector. However, any noise that is observed in the output measurements can be added to the state equation in the form of its variance. By accounting for some of the noise in the data, the Kalman filter will not add it into the disturbance estimate, which will improve the results. In this case, the quantization error caused by the encoder is considered

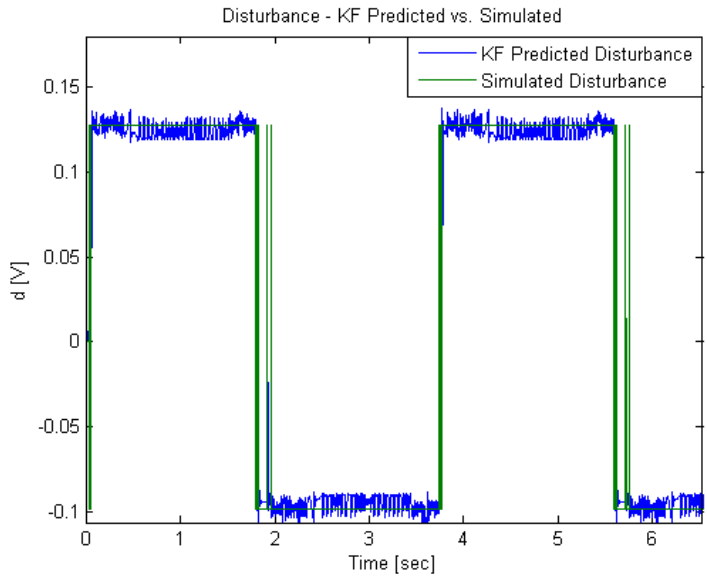


Figure 4.9: Kalman Filter Disturbance Prediction - Variance = 0.008

and represented as R_v, with a zero-mean Gaussian distribution. Equation 4.17 shows the modified discrete time state space model, which includes the disturbance dynamics.

$$\begin{aligned} \underbrace{\begin{bmatrix} x(k+1) \\ d(k+1) \end{bmatrix}}_{\dot{z}} &= \underbrace{\begin{bmatrix} A & -B \\ 0 & 1 \end{bmatrix}}_{A_Z} \underbrace{\begin{bmatrix} x(k) \\ d(k) \end{bmatrix}}_{z} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B_z} \begin{bmatrix} u(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{W} \begin{bmatrix} \omega_d(k) \end{bmatrix} \\ \begin{bmatrix} y_z(k) \end{bmatrix} &= \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{C_z} \underbrace{\begin{bmatrix} x(k) \\ d(k) \end{bmatrix}}_{z} + R_v \end{aligned} \tag{4.17}$$

The Kalman filter gain can then be determined using the combined model form of (4.17) with the set of gain equations in 4.18 below.

$$P(k|k-1) = A \cdot P(k-1|k-1) \cdot A^T + W \cdot R_w \cdot W^T \tag{4.18a}$$

$$K_{obs} = P(k|k-1) \cdot C^T [R_v + C \cdot P(k|k-1) C^T]^{-1} \tag{4.18b}$$

$$P(k|k) = (I - K_{obs} C) \cdot P(k|k-1) \tag{4.18c}$$

These equations use the system state matrices and variances to estimate the error covariance ($P(k|k)$) of the Kalman filter state estimate (equation 4.18a). Equation 4.18b uses

the estimate of the error covariance to calculate the Kalman gain term (K_{obs}). The Kalman gain is then used in equation 4.18c to improve the estimate of the error covariance. By iteratively repeating this process, the Kalman gain will eventually converge to a final value, determined by a convergence tolerance.

Finally, with the modified model form in equation 4.17 and the calculated Kalman gain, the standard form of the Kalman filter can be used to estimate the state vector. Equation 4.19 contains the Kalman filter equation, using the model state space equations defined above. The Kalman filter designed here will estimate the disturbances experienced by the system during any motion. The benefit with this form of Kalman filter is that it can be run offline. The input and output data of an experimental trial must only be recorded and can then be analyzed later. This is an advantage when tuning the parameters or structure of a model, as new experiments do not need to be run, which saves time.

$$\hat{z}(h) = (I - K_{obs} \cdot C_z)A_z \cdot \hat{z}(k-1) + (I - K_{obs} \cdot C_z)B_z \cdot u(k-1) + K_{obs} \cdot y_z(k) \quad (4.19)$$

4.4.2 Friction Model Identification

The Kalman filter described in section 4.4.1 must be used in conjunction with smooth velocity tests. The main principle is that any fluctuations seen during a constant velocity test will be attributed to disturbance. When the Kalman filter estimates the disturbance profile during these tests, it is estimating the value of friction at different speeds. These tests cannot be run in an open-loop configuration. To ensure that the desired velocity is reached and maintained during the trials, a simple proportional-derivative (PD) controller was developed. This controller was designed using the identified parameters for the system, shown in table 4.3. The criteria selected for the controller was a rise time (t_r) of 0.05 seconds and a maximum overshoot (M_p) of 10 percent. Using the standard second order system transfer function expression, equation 4.20 below, the gains for the PD controller were calculated using standard methods [9] and are presented in table 4.5. A block diagram of the system is presented in figure 4.10.

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n \cdot s + \omega_n^2} \quad (4.20)$$

The PD controller achieves satisfactory position control. By inputting a linearly increasing position command, the system is able to reach a constant velocity. This controller

Gain	Value
K_P	1.1466
K_D	0.0363

Table 4.5: PD Controller Parameters

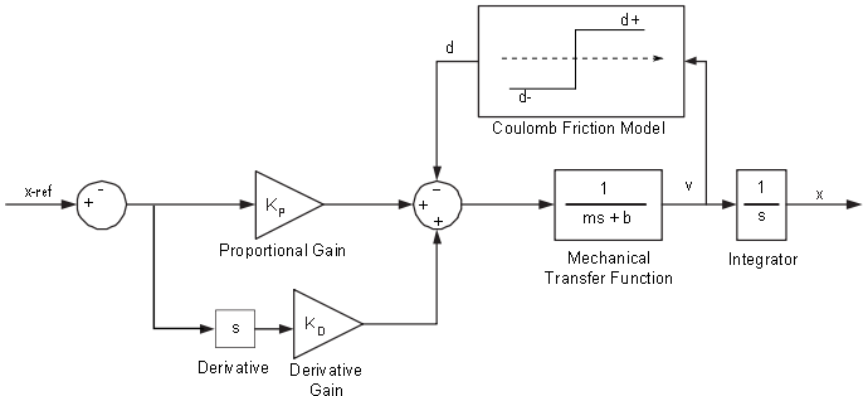


Figure 4.10: System Block Diagram with PD Control

enables the velocity trials for friction parameter identification to be run, but it is not the final controller used for machining. The controller used for machining is presented in Section 5. Figure 4.11 presents a sample position command for the smooth velocity tests.

As can be seen in the figure, the constant slope of the position curve will achieve a constant velocity. The command profile will move the system both forwards and backwards, achieving the necessary positive and negative velocity measurements in one test. Generally, the Kalman filter will produce a better disturbance estimate when more data is collected. However, the distance the x-axis is able to travel in either direction is limited by the length of the ball screw and the placement of the bearings on the slide rails. Starting at the rear position of the ballscrew, the platform can move forward approximately $190mm$. Therefore, the distance of travel for each trial is kept at a conservative constant, while the duration of each trial is reduced as faster velocities are tested. All of the velocities that were tested are listed in table 4.6 below.

The Kalman filter presented in section 4.4.1 can now estimate the disturbances in the system at different velocities by using the collected data. The two figures in 4.12 are typical results, showing the estimated disturbance for two sample velocity trails. From

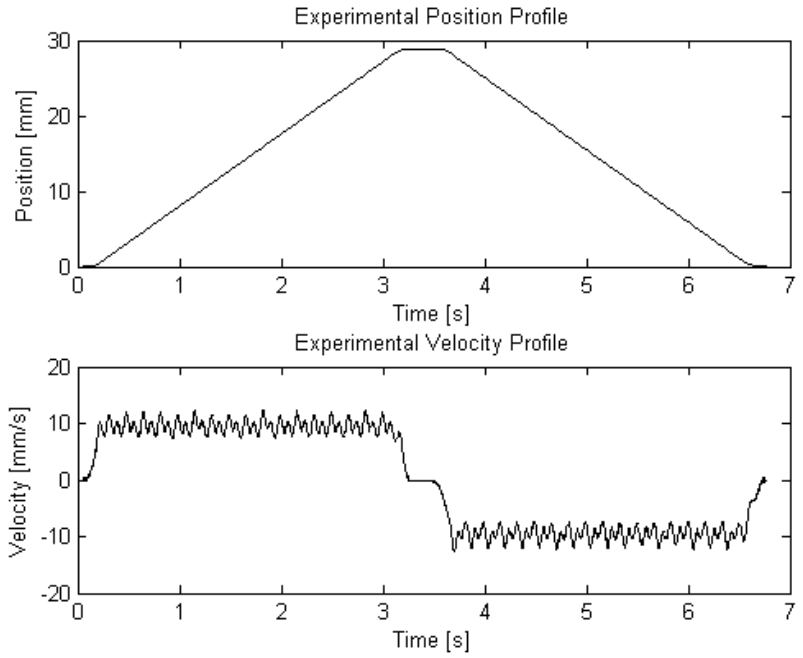


Figure 4.11: Example Smooth Velocity Trial for ±10 [mm/s]

±0.1	±0.25	±0.50	±0.75	±1.0	±2.0	±3.0	±4.0	±5.0
±6.0	±7.0	±8.0	±9.0	±10.0	±25.0	±50.0	±75.0	±100.0

Table 4.6: Smooth Velocity Trial Values - In [mm/s]

these disturbance estimates, an average disturbance value is calculated for each test. The average disturbance value for the two sample trials are shown in the figures as the constant straight lines running through the estimated disturbance data.

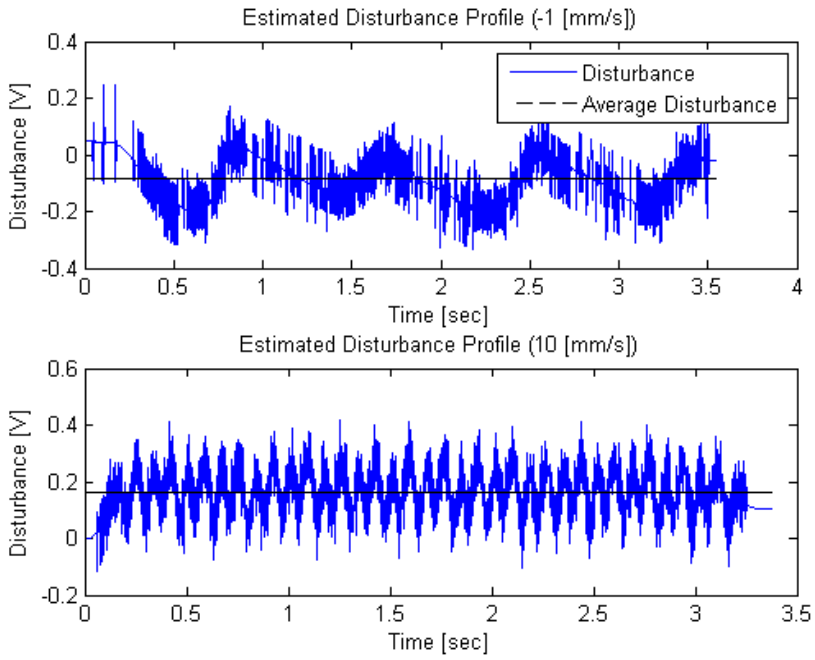


Figure 4.12: Example Disturbance Estimates for Two Test Velocities

By plotting the average disturbances against their respective constant velocities, the disturbance plot can be generated. As can be seen in figure 4.13, each point in the plot represents an average disturbance estimated by the Kalman filter at the corresponding velocities. This disturbance profile looks similar to the Stribeck curve shown in figure 4.3. The profile shape further validates the earlier assumption to attribute the majority of the observed disturbance to friction. As indicated in the Stribeck curve, a slope in the disturbance values as the velocity increases in magnitude in either direction is attributed to bias in the viscous damping parameter. By correcting the value of viscous damping, the slope in the disturbance profile will appear more horizontal as it moves towards positive and negative infinity, which removes the bias. To correct the viscous friction term, the slope of the disturbance estimate in the positive and negative directions is averaged and then added onto the viscous friction parameter. For the disturbance curve in figure 4.13, the approximate average slope is found to be $-0.000633\,V/\frac{mm}{sec}$. This slope is added to the

viscous friction parameter b, which are of the same units, and the disturbance estimate is calculated again using the Kalman filter and the same experimental data, but now with the new value of $b = 0.001944$. Figure 4.14 shows the second iteration of the disturbance estimate profile. This profile exhibits a slope that is closer to zero as the curve moves towards positive and negative infinity than the profile in figure 4.13. This indicates that the tuned value of b has a smaller amount of bias than the previous value and is a more accurate estimate for the system.

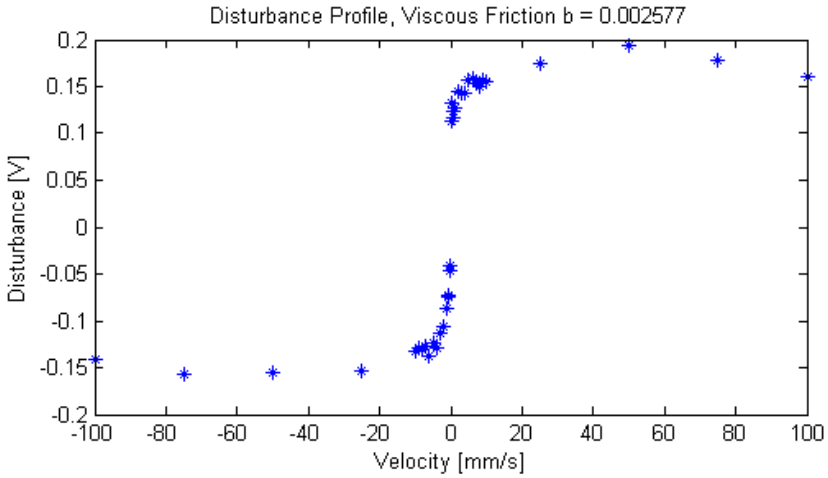


Figure 4.13: Estimated Disturbance Profile Using Kalman Filter

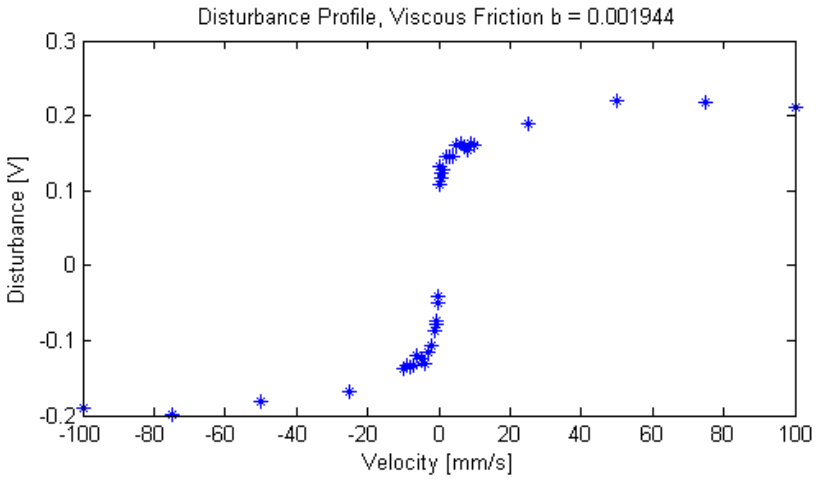


Figure 4.14: Tuned Disturbance Profile

The estimated disturbance profile can be used to determine the corresponding static and

coulomb friction values that are required for equation 4.12. Using figure 4.14, the static friction terms are identified based on the disturbance value estimated for the smallest velocity magnitude for both the positive and negative directions. The coulomb friction terms were estimated at where the disturbance values leveled off in the positive and negative velocity directions, as seen in the Stribeck curve. To find an estimate for the velocity constant Ω, a cost function,

$$\Delta = \frac{1}{2} \cdot \sum \delta^2, \tag{4.21}$$

was used, where δ is the difference between the disturbance given by the friction model and disturbance estimate at a velocity point. Equation 4.21 penalizes the error between the experimentally determined disturbance and the disturbance calculated using the model. Values of Ω were substituted into the friction model from 0.1 to 10 at steps of 0.1. The value of Ω that minimized the cost function was selected as the estimate for the model. The friction model parameters are presented in table 4.7.

Parameter	**Value** $[V]$
(d_s^+)	0.109
(d_s^-)	-0.0496
(d_c^+)	0.2095
(d_c^-)	-0.1847

Table 4.7: Nonlinear Friction Model Parameters

To verify the identified nonlinear friction model parameters, the friction model was plotted against the estimated disturbance profile that was found using the Kalman filter. As can be seen in figure 4.15, the friction model predicts the higher velocity points accurately, but does not match the lower velocity portion of the curve as nicely. This is due to the exponential in the friction model, equation 4.12, causing a smooth curve up to a constant value. The approximation is acceptable because the majority of the intended motion of the x-axis is composed of velocity values in the higher region, where the curve is more accurate. For any errors incurred by the disturbance function at lower velocities, the robustness of the controller will be relied upon to compensate accordingly.

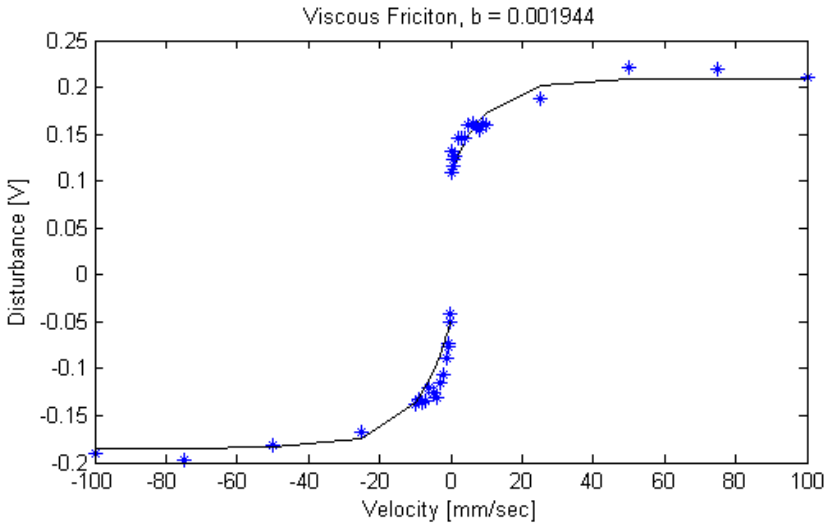


Figure 4.15: Estimated Disturbance Profile vs. Friction Model

4.5 Model Validation

The nonlinear friction model developed in section 4.4 will replace the simple coulomb friction model that was identified along with the inertia and viscous damping parameters in section 4.3. To validate the friction model, the input signal shown in figure 4.4 is applied to the new system model. Figure 4.16 shows the results. As can be seen in the results, the system with the new friction model appears to closely follow the experimental output data. To compare this result to that of the simpler coulomb friction model developed earlier, an error plot was created based on the differences between the experimental and simulated output for both friction models. Figure 4.17 illustrates this comparison. The figure indicates that both friction models provide similar estimates for higher velocity inputs. This is expected, as both models contain an estimate for coulomb friction, which is the dominant friction term at higher velocities. However, during slower velocity motions, the nonlinear friction model produces a significantly smaller amount of error than that of the coulomb friction model. Again, this is expected because the nonlinear friction model takes into account a static friction estimate, which is the dominant friction term during slow velocity motion. For example, during the time interval from 0.8 to 0.9 seconds, which is a higher velocity motion, the approximate error values are 3.42 and 2.96 mm/s for the nonlinear and coulomb friction models respectively. This is a difference in error of about

0.46 mm/s. During the time interval of 0.2 to 0.3 seconds, a slower velocity motion, the approximate error values are 4.73 and 2.57 mm/s for the nonlinear and coulomb friction models respectively. This is a difference in error of about 2.16 mm/s. These results indicate that the nonlinear friction model, which takes into account both static and coulomb friction terms, provides a more accurate model than the simpler coulomb friction model. With the addition of this friction model, the model for the x-axis of the CNC lathe is now complete.

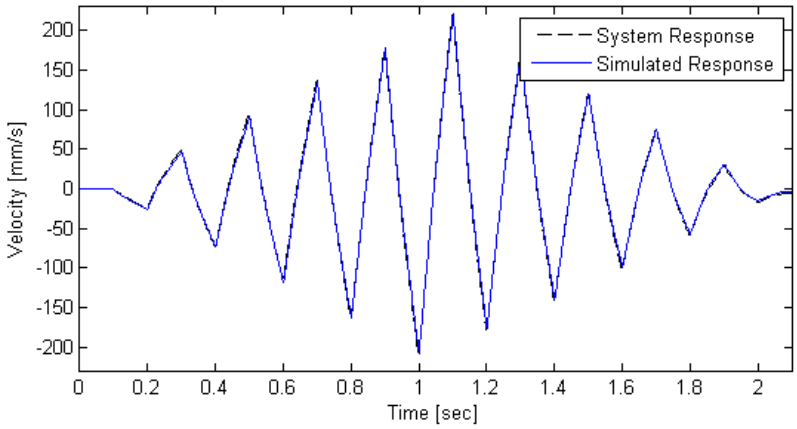


Figure 4.16: Nonlinear Friction Model Validation

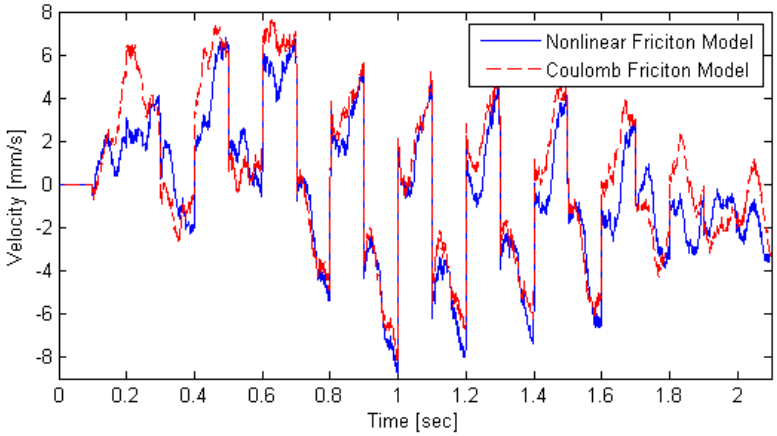


Figure 4.17: Friction Model Error Comparison, Nonlinear vs. Coulomb

4.6 Summary

The final model for the x-axis of the 3-axis lathe is presented in figure 4.18. This model contains a first order transfer function that has been simplified into controller equivalent signals and parameters. The parameters were identified using a variation of the least squares parameter identification technique. A nonlinear friction model was developed using a Kalman filter to estimate the disturbance injected into the system that considered both static and coulomb friction. Developing the friction model also improved the estimate of the model's viscous damping parameter. By adding this friction model to the x-axis model, the accuracy of the simulations of the system was improved. This model will be used to develop and tune a control law for operating the lathe at faster speeds. Having an accurate model will allow this process to be more effective and reliable.

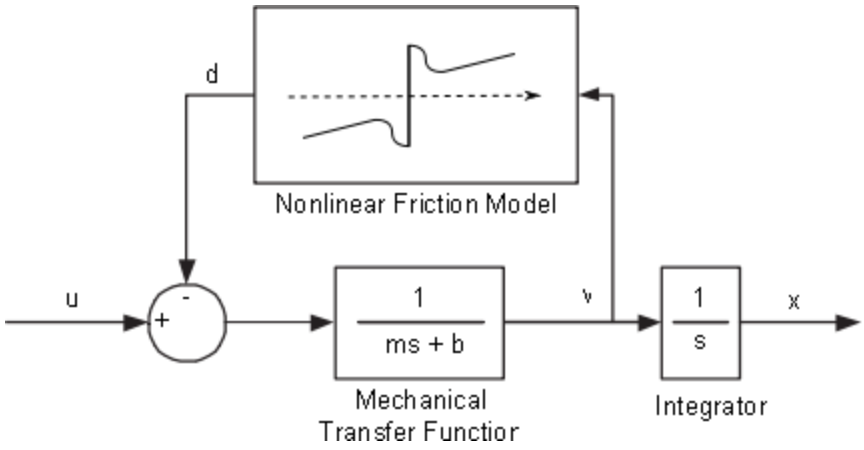


Figure 4.18: System Block Diagram with Nonlinear Friction Model

Chapter 5

Controller

5.1 Controller Selection

The controller selected for use with the 3-axis CNC lathe system was an adaptive sliding mode controller (ASMC). The decision was made based on the expectation of the performance of the controller on the mechanical system, and on the ease of its implementation. The system is required to track position commands with an error of less than $0.5mm$. This error threshold comes from the fact that in a wood surface, errors greater than $0.5mm$ will show-up visually. The ASMC method holds numerous attractive qualities that make it a strong choice for a control law. Firstly, the performance of the ASMC has been shown to be good in terms of tracking error and robustness in other applications, such as in electric drives for machine tools, vehicle control and mobile robot control [3, 22]. This is due to the controller having a high bandwidth and an inherent feedforward design, coupled with disturbance rejection. In fact, the ASMC has been shown to be less sensitive to changes in the mechanical system model than other conventional and complex control laws alike [2].

Second, the ASMC is also a relatively easy control law to design and tune. There are only two main steps to the design of a sliding mode controller; the selection of a sliding surface and a Lyapunov function. Normally, the ASMC is a non-linear control law that is robust to uncertainties and time variation in the system plant or external forces [2]. However, if there is only one sliding surface parameter, disturbance rejection for example, the control law works out to be linear, while still retaining its robust properties [4]. As

the development of the ASMC in the following section will discuss, disturbance rejection was selected as the sliding surface parameter for the control law.

Altintas et al. [2] performed a direct comparison between an ASMC and a pole placement controller with feed forward action and a zero phase error tracking (ZPET) filter. It was found that both controllers performed equally well in terms of tracking error for specific machining profiles, but the ASMC required much less design and fewer computations. The ASMC used a simpler model and was able to reject disturbances better than the ZPET controller. This is also true in the case of more conventional controllers, such as PID controllers, where the system does not achieve high tracking accuracy at high speed if there are too many external factors (like friction) that are either not modelled or not compensated for appropriately.

One of the drawbacks with using a sliding mode controller is the choice of the sliding surface and Lyapunov function must be made based on experience and intuition. This implies that a solution to the control law may not be an optimal solution. Also, it has been shown that the ASMC can produce overshoot and transient oscillations if the frequency content of the commanded input contains values that are outside of the controller's bandwidth. Despite this small downside to the ASMC, it was selected as the control law to run the 3-axis CNC lathe system, based on its tracking performance and ease of implementation.

5.2 Control Law

The theory that follows for the ASMC control law was adapted from [4, 22]. The control law for the ASMC is created based on the model presented by equation 4.1 in section 4.2. This model was first rewritten in state space form to more easily create the ASMC design. The state space model is presented as

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-b}{m} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t) - \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} d(t). \tag{5.1}$$

Next, a sliding surface must be selected. Due to the model being second order, the sliding surface defines a line in the state space of the system, instead of an actual surface. The ASMC will drive the state of the system to this line and once on the sliding surface, the controller will keep the state there. For the system to be well behaved, the state must

move towards an equilibrium point of zero, while remaining confined to the surface. This equilibrium point is defined for x approaching x_{ref}, or when the error is zero. Hence, by transforming the state of the system into the error of the state and moving that error towards zero, position control is achieved. The sliding surface definition is

$$\sigma = S\left(\begin{bmatrix} x_{ref} \\ \dot{x}_{ref} \end{bmatrix} - \begin{bmatrix} x \\ \dot{x} \end{bmatrix}\right). \tag{5.2}$$

The term S in equation 5.2 is a matrix that represents the constants in the sliding surface equation that force the equation to zero when $\sigma = 0$. For simplicity, one of the constants is set to one, while the other is represented by λ. This constant can be related to the bandwidth of the controller and must be selected based on the capabilities of the physical system and the intended input commands.

$$\sigma = \underbrace{\begin{bmatrix} \lambda & 1 \end{bmatrix}}_{S}\left(\underbrace{\begin{bmatrix} x_{ref} \\ \dot{x}_{ref} \end{bmatrix}}_{x_{ref}} - \underbrace{\begin{bmatrix} x \\ \dot{x} \end{bmatrix}}_{x}\right) \tag{5.3}$$

To ensure that the state remains on σ, the time derivative of σ evaluated at the state trajectories must be zero, or, $\dot{\sigma} = 0$. From equation 5.3 this becomes

$$\dot{\sigma} = S\left(\dot{x}_{ref} - \dot{x}\right). \tag{5.4}$$

Using the system model, equation 5.1, the expression becomes

$$\dot{\sigma} = S\left(\dot{x}_{ref} - Ax - d\right) - SBu(t). \tag{5.5}$$

To make this equation zero and satisfy the stability and performance criteria, the control law u can be set to

$$u = (SB)^{-1}S(\dot{x}_{ref} - Ax + Bd). \tag{5.6}$$

The above control law will produce a system response that has the desired position tracking performance. However, this is only true assuming all of the parameters in the model presented in figure 4.18 are exact. To make the controller more robust, the control law can be modified to account for a degree of uncertainty in some or all of the model parameters. The parameters in question are the system's mass m, viscous friction b, and disturbance

d. The disturbance is considered as an uncertain parameter. This is an appropriate choice because the disturbance, which has been treated wholly as friction, is difficult to model. It is also possible that other unmodelled dynamic effects could be present in the disturbance parameter. Also, any error that exists in the other estimated system parameters will be represented to a certain extent in the disturbance.

First, equation 5.6 can be rewritten as a linear combination of r parameters $P = \begin{bmatrix} p_1 & p_2 & \cdots & p_r \end{bmatrix}$ and corresponding regressors $Y(\dot{x}_{ref}, x) = \begin{bmatrix} y_1 & y_2 & \cdots & y_r \end{bmatrix}$, represented as $Y(\dot{x}_{ref}, x)P$. By expanding equation 5.6 and putting it into this form, we get

$$u = \underbrace{\begin{bmatrix} \ddot{x}_{ref} + \lambda(\dot{x}_{ref} - x) & \dot{x} & 1 \end{bmatrix}}_{Y} \underbrace{\begin{bmatrix} m \\ b \\ d \end{bmatrix}}_{P}. \tag{5.7}$$

The uncertainty in the disturbance parameter is considered to be bound by $d^- \leq d \leq d^+$ and represented by the equation

$$\dot{\hat{d}} = \rho\kappa y_3 \sigma, \tag{5.8}$$

where $\dot{\hat{d}}$ is equal to an estimate of the disturbance. The disturbance equation was constructed in the following manner. The term ρ represents the adaptation gain, which is the sensitivity of the controller to change. The adaptation bound factor κ, controls when the parameter needs to change. If the disturbance is outside its bounds, κ will be equal to one, representing the need for the parameter to change, if the disturbance is within its bounds, κ will be zero. The variable y_3 controls how much to change each parameter by. From equation 5.7, $y_3 = 1$.

A control law that guarantees the stability of this new form is found using a Lyapunov approach. A Lyapunov function $V(x)$, for equilibrium point $x = 0$, must be found that increases in value with the system states, or their error and satisfies the following criteria,

$$\begin{aligned} &\bullet\, V(x) > 0 \\ &\bullet\, \dot{V}(x) \leq 0, \text{ for system stability} \\ &\bullet\, \dot{V}(x) < 0, \text{ for system asymptotic stability.} \end{aligned} \tag{5.9}$$

By constructing a control law that allows the Lyapunov equation to satisfy all of the conditions above, the system will be asymptotically stable. This implies that not only will

the states of the system be bounded, but that they will approach the equilibrium point of the system in a finite amount of time. This will guarantee the robustness of the control law, even when faced with uncertainties in the disturbance parameters. A Lyapunov function that satisfies the stated criteria was found in [4] and is presented below:

$$V(x) = \frac{1}{2}[\sigma^T (SB)^{-1}\sigma + (P - \hat{P})^T \Gamma (P - \hat{P})]. \tag{5.10}$$

In this general form of the Lyaponov function, the term $\hat{P}$ is identical in structure to the matrix P; however all of the variables of the matrix are considered to be perturbed versions of the parameters in P, within a known bound. As mentioned above, the control law is being designed so that only the disturbance d is considered to have any perturbations. This implies that $m = \hat{m}$ and $b = \hat{b}$. Similarly, the term Γ is a diagonal matrix that contains the adaptive gains ρ_r. Only the gain for the disturbance parameter is a variable however, represented as ρ_3 in equation 5.8. The remaining gains are set to a value of 1, since the mass (m) and viscous friction (b) parameters are not considered to be uncertain. The Lie derivative of equation 5.10 can be simplified as

$$\dot{V}(x) = (P - \hat{P})Y^T\sigma - (P - \hat{P})^T \Gamma \dot{\hat{P}} - \sigma^T u + \sigma^T Y \hat{P}. \tag{5.11}$$

It has been proven that to ensure equations 5.10 and 5.11 satisfy conditions 5.9, a suitable control law is

$$u = K_s\sigma + Y(\dot{x}_{ref}, x)\hat{P}, \tag{5.12}$$

where K_s is greater than zero.

Given this control law, we can now simplify it into a form that can be implemented on the x-axis model of the system. Using equation 5.12 and determining the integral of the uncertain disturbance parameter as

$$\hat{d} = \rho(x_{ref} - x) + \rho\lambda \int_0^t (x_{ref} - x)\, d\tau, \tag{5.13}$$

the control law can be represented as

$$u = \hat{m}\ddot{x}_{ref} + \hat{b}\dot{x}_{ref} + [\underbrace{K_s\lambda + \rho}_{K_P}]e + [\underbrace{\ \rho\lambda\ }_{K_I}] \int_0^t e\, d\tau + [\underbrace{K_s + \hat{m}\lambda - \hat{b}}_{K_D}]\dot{e}. \tag{5.14}$$

This control law will guarantee the tracking performance of the x-axis, while also remaining robust in the face of the uncertainties expected in the disturbance parameter. Once

designed, this control law is relatively easy to implement, only requiring the tuning of three parameters. As mentioned above, the tracking performance of the ASMC is comparable to other known controllers. Figure 5.1 shows a block diagram of the ASMC controller.

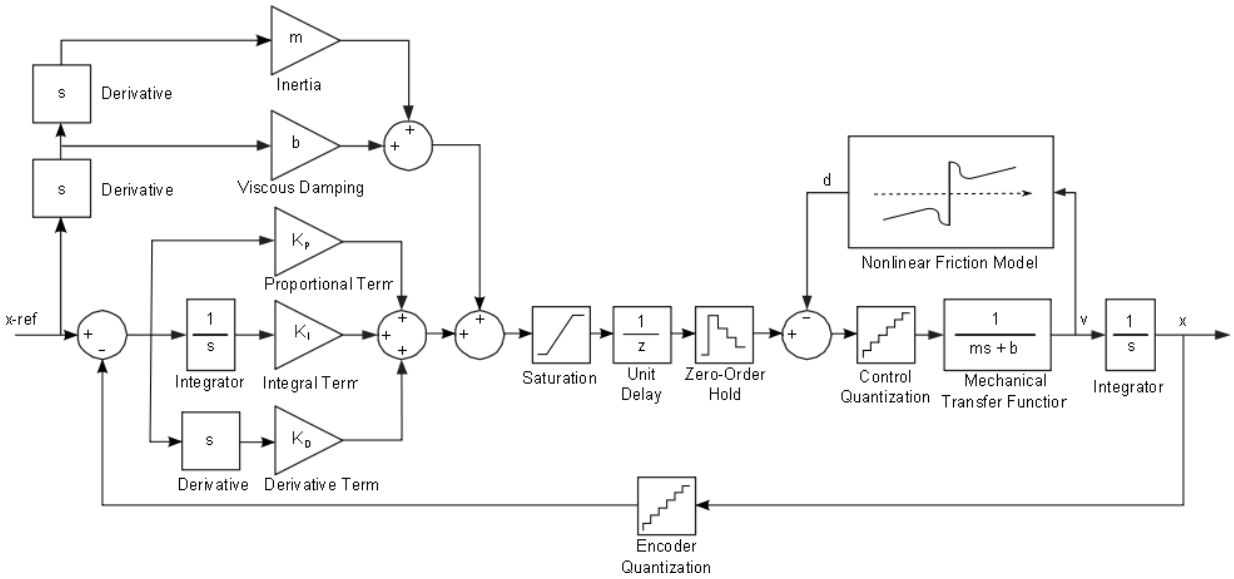


Figure 5.1: Adaptive Sliding Mode Controller Block Diagram

5.3 Parameter Tuning

The parameters that require tuning are λ, the sliding surface parameter that forces the systems states towards zero and represents the bandwidth of the system; ρ, the adaptation gain, controlling the sensitivity of the controller to uncertainties; and K_s, the controller law gain that is used to satisfy the Lyapunov function.

The parameter λ is selected by choosing a bandwidth for the controller. To ensure that the bandwidth selected is not higher than that of the system's, a frequency response test was run on the system. A frequency response test is performed by inputting a series of sine waves to the system and measuring the response to generate a bode plot of the system. The bode plot, which contains a magnitude and frequency plot of the system's response, will indicate an experimental bandwidth drop-off point, which can be used as the bandwidth limit of the system. The results from the frequency response test are shown in figure 5.2.

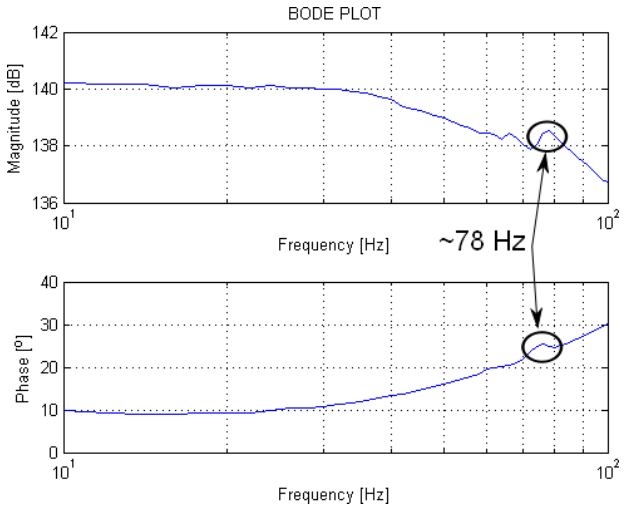


Figure 5.2: X-Axis Frequency Response

As can be seen in the figure, a small resonance peak occurs in both the magnitude and phase plots at approximately 78Hz. Operating the system within the range of a resonance peak can have negative effects, such as extreme vibrations, and should be avoided. Using 80Hz as an estimate of the system's bandwidth, the controller bandwidth is selected to be 40Hz; well away from the bandwidth of the system. Selecting a lower bandwidth for the controller should ensure that the system will not enter the resonance peak during operation and should also avoid .

Once λ has been selected appropriately for the given system, the remaining two parameters ρ and K_s are selected through simulation. Using the system model and controller the tracking error of the system is simulated for different values of ρ and K_s. A range of ρ from 0 to 10 in steps of 1, and a range of K_s from 0.01 to 0.31 in steps of 0.01 is used. By minimizing the sum of the square of the tracking error for the different trials, a ρ and K_s can be selected for optimal tracking error performance. Table 5.1 below presents the tuned parameters that were used with the adaptive sliding mode controller.

5.4 Validation

To validate the performance of the controller on the x-axis of the lathe system, the controller was simulated on the model provided in equation 4.1. A smooth trajectory, shown in figure

Table 5.1: Tuned ASMC Parameter Values

Variable	Value
λ	40
ρ	0.04
K_s	10

5.3, was used as the test input. The travel distance of this trajectory is $150mm$, which is the maximum distance the x-axis system can move forward safely. The feedrate was chosen to be $100mm/s$, which should be an easily attainable speed for a smooth motion profile. This trajectory, which is designed to be jerk continuous, will test the ability of the controller to achieve a desired position tracking error performance. The results of the simulation are shown in figure 5.4.

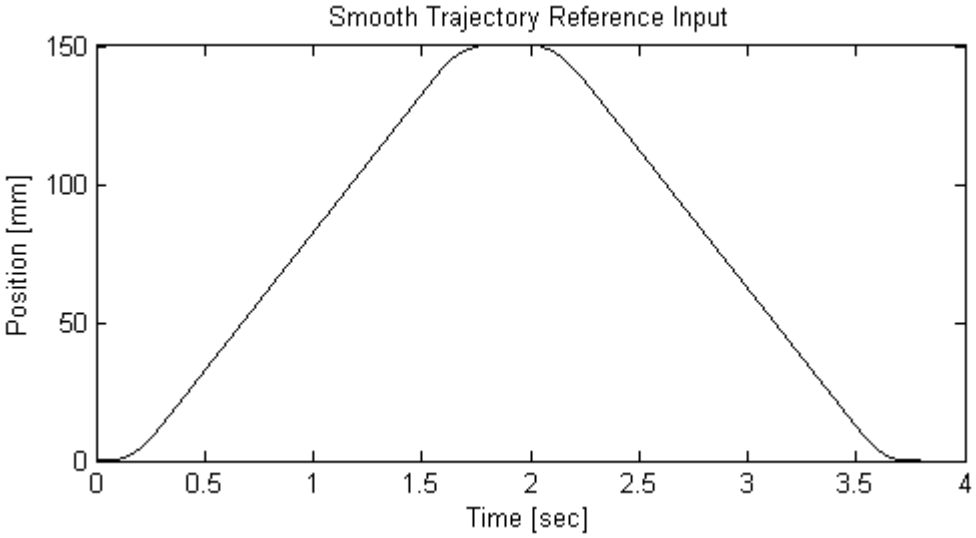


Figure 5.3: Smooth Trajectory Reference Input

As can be seen in the figure, the ASMC is able to achieve a maximum position tracking error of $0.023mm$. This value is well below the desired maximum of $0.5mm$, which is discussed in section 7. To further validate the performance of the sliding mode controller, the control law is implemented on the physical system and the same input is used to test the performance. Figure 5.5 shows a comparison between the results of the simulated

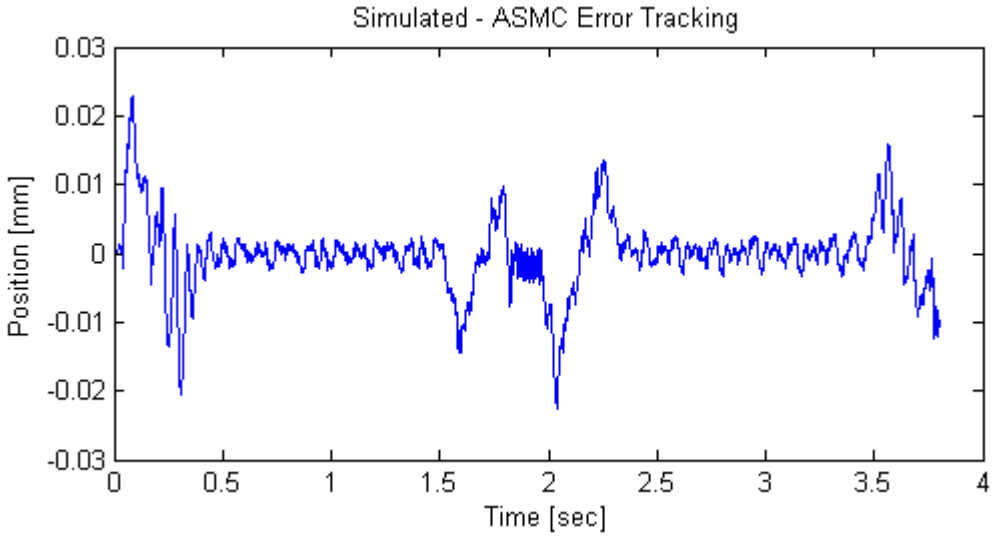


Figure 5.4: Simulated ASMC Tracking Error

ASMC and experimental ASMC. The experimental results validate those of the simulation, which follow the same trends and are within the same magnitude range. The experimental error results match closely to the simulated during the constant velocity intervals, but the results differ greatest during the velocity transitions. Somewhat uncommonly, the simulated response contains the larger error during these transitions, reaching magnitudes that are approximately twice as large as those seen in the experimental response. These differences can be explained by errors in the identified model. The largest discrepancies between the simulated and experimental results are likely due to the friction model that is being used, since these differences occur during the velocity transitions. An accurate friction model is difficult to create and can easily influence the behaviour of a system at low velocities, where the friction terms have a greater effect. It is also known that the friction model being used is in the system model is less accurate at lower velocities, due to the approximations made during model development (Section 4.4). However, the results of the simulated and experimental trials are similar enough for the purpose of this work. More importantly, they validate the performance of the sliding mode controller.

To further demonstrate the ASMC's ability to reach high levels of position tracking performance, the experimental results of the controller are compared to the results of a PD controller. The PD controller used for comparison is outlined in section 4.4.2 that was used for controlling the motion of the x-axis during the friction identification experiments.

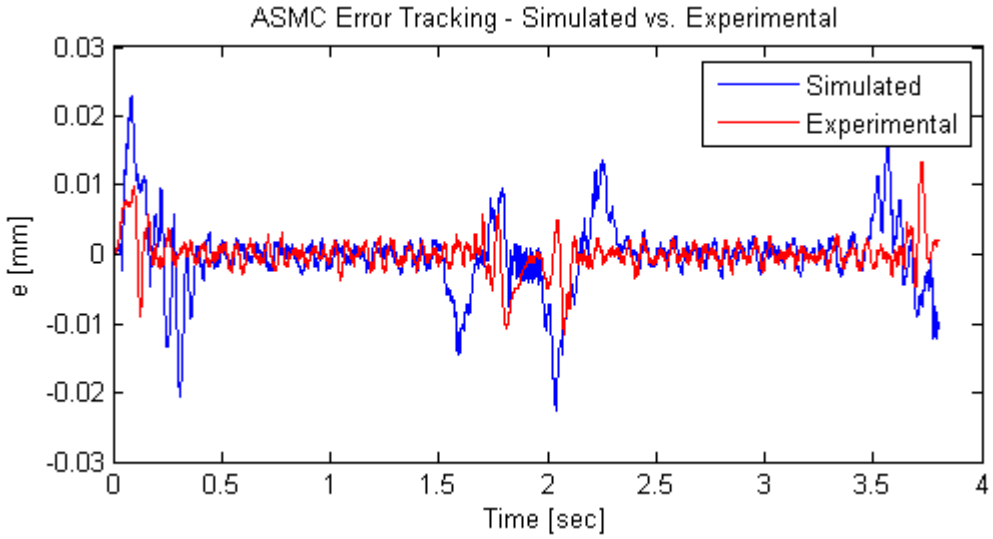


Figure 5.5: Simulated vs. Experimental ASMC Tracking Error

Figure 5.6 shows the comparison between the experimental results for the two controllers.

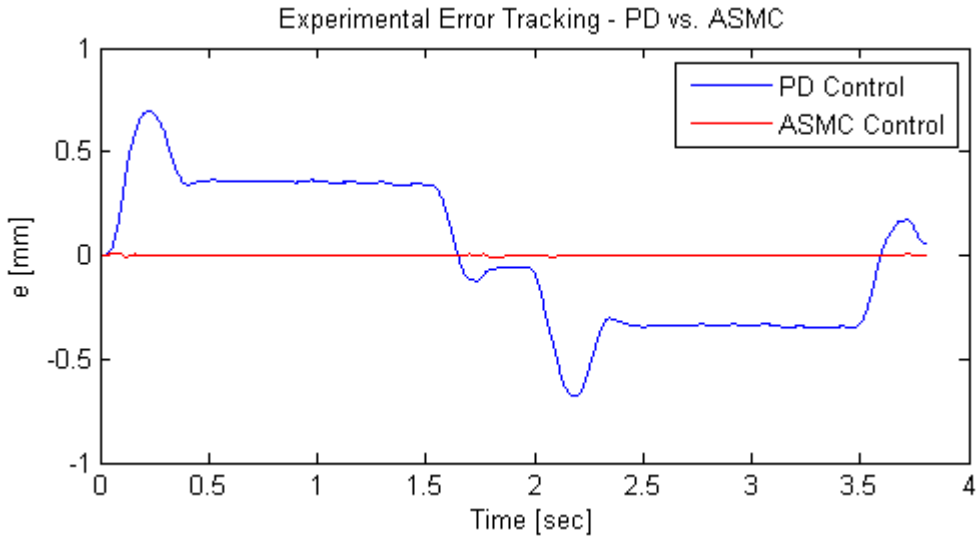


Figure 5.6: PD vs. ASMC - Experimental Tracking Error

As the results in figure 5.6 clearly illustrate, the position tracking error generated by the ASMC is much smaller than that of the PD controller for the same input signal. The PD controller has a maximum error of approximately $0.69mm$, which is approximately 30 times larger than the ASMC's $0.023mm$ of tracking error.

The control law presented in this section has been shown to provide acceptable position tracking performance through experimental validation. The performance of the sliding mode controller has been compared to that of a PD controller and has been shown to achieve better tracking performance, reinforcing the decision to make the ASMC the controller for the 3-axis CNC lathe system. The control law presented in this section also shows that the ASMC is robust to an expected amount of uncertainty in the disturbance parameters in the model. This allows the ASMC to reliably control the x axis of the system when unexpected disturbances are introduced into the system. These disturbances could include unmodelled friction effects, such as wood chips on the ball screw, or external resistance forces generated from cutting the material.

Chapter 6

Algorithms

The 3-axis CNC lathe system uses a number of unique software packages in its operation. The system makes use of a custom toolpath planner that was specifically designed for the lathe system. A trajectory generator takes the toolpath and creates a smooth trajectory with feed profile that is jerk continuous. This trajectory is used to send commands to the three motors of the system. An algorithm is also proposed that takes a toolpath and smooths the geometry of sharp edges automatically. This algorithm can be used for smoothing edges of a model by a minimum amount to achieve a desired speed.

6.1 Toolpath Planner

The toolpath planner used for the 3-axis CNC lathe system is based on the works of Manos [15] and Patel [20]. The program requires a model in stereo lithography (STL) file format, which is a file that describes a solid model using only triangles to approximate its surface. The toolpath planner uses a method called the 'ball drop' approach. A sphere is used to approximate the shape of a ball-nose end mill and is 'dropped' onto the triangulated model surface. The first point of contact between the sphere and the model surface is taken as the toolpath point.

To calculate this point, all of the triangles in the 'shadow' of the sphere are located. The shadow of the sphere is considered to be a projected circle of the sphere onto the model surface. The shadow would be the portion of the model that is removed if the ball-

nosed cutter were to plunge straight into the surface. Next, considering only the triangles that lie in the shadow of the sphere, a point of contact is determined. For each triangle the algorithm checks for the point of contact between the sphere and the plane in which the triangle lies, the line segments that form the triangle, and the vertices of the triangle. From these points of contact, the point that is furthest from the axis of rotation of the model is selected. In this way, the ball-nose end mill will move towards the model surface and stop at the determined point of contact, ensuring the surface is free of gouging. The toolpath is created by rotating the sphere around the model at regular predefined intervals and performing the 'ball drop' calculation. This forms a helical toolpath around the model that covers the entire surface, as described in section 3.

6.2 Trajectory Generator

The purpose of the trajectory generator is to take the toolpath points from the algorithm described above and create a set of intermediary points for the motors to follow that will provide a smooth motion for the tool. The term 'smooth motion' is being used to imply finite velocity, acceleration and jerk values. The jerk profile of the tool motion will not only be of finite value, but will also be continuous. The algorithm presented here was developed in [4, 1] and only needed small modifications to be adopted to the toolpaths being used in this work. The main modification was a change in how the algorithm interacted with the data. Instead of calculating the entire trajectory at once, the toolpath needed to be broken down into smaller sections. This was required because the toolpaths being dealt with contained far too many points for the solution to be computed in the PC memory. This was the case because the solution calls for the inversion of a square matrix, whose sides are four times the number of data points. For several thousand toolpath points, the amount of memory to hold the square matrix could reach the thousand's of gigabytes. Each small solution of the trajectory therefore had to be stitched back together to match the full motion of the toolpath. This stitching did not appear to have an effect on the smoothness of the trajectories.

This method uses a second-order curve to describe the jerk profile of the desired tool motion and consequently develops into a fifth-order position profile for the position of the intermediary toolpath points. The benefit of having a continuous jerk and therefore

continuous position, velocity and acceleration profiles, is that the demands on the motors propelling the system are reduced. Smooth trajectory profiles lead to better performance from the motors, allowing them to go faster, as they are more easily able to meet the dynamics that are being demanded. This creates less jitter in the motion of the tool, creating a better surface finish and makes the system easier to control, improving the accuracy. Along with this smooth trajectory profile, the algorithm uses a cubic spline to approximate a smooth curve between the toolpath points that have already been provided. The trajectory generating algorithm can be broken up into the following steps:

1. Solve the parameters for the cubic splines between each pair of toolpath points.
2. Calculate the total arc length of the splines.
3. Calculate the smooth feed profile of the motion.
4. Interpolate the smooth intermediary toolpath points.

Once the trajectory algorithm has completed, the resulting motion profiles for each axis will be smooth and constructed to match the sampling frequency of the system. This motion profile is what is passed to the controller of the 3-axis CNC lathe system during machining.

6.2.1 Cubic Spline Parameters

A cubic spline is used to create a smooth curve between each toolpath point that describes a solid model. In this way, the surface of the model is smoother than when approximating the surface with straight lines between toolpath points. A unique spline must be solved for each pair of toolpath points. To make sure the separate splines create a smooth curve, their parameters must be solved together, using a set of continuity equations. First, the chord length l_k between each toolpath point P_k must be calculated. As depicted in figure 6.1, the chord length is the distance between a set of two toolpath points and can be calculated much like the magnitude of a vector.

$$l_k = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2 + (z_{k+1} - z_k)^2}. \tag{6.1}$$

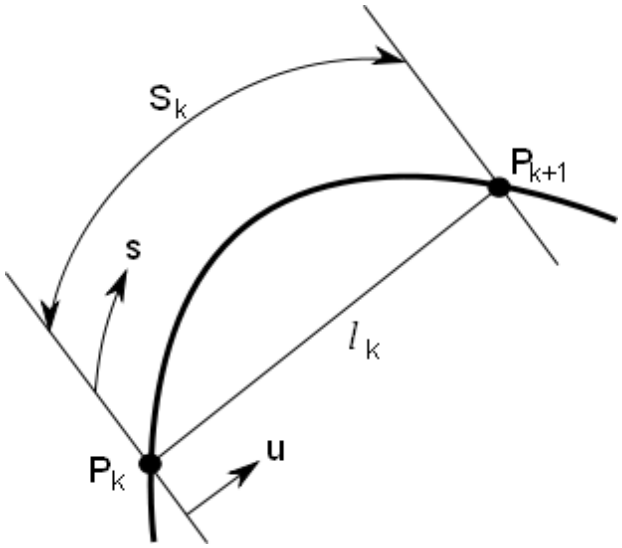


Figure 6.1: Chord Length Calculation

Using the chord length distances, a unique set of splines can be calculated for each axis separately. The following equation is the general form used for the cubic spline.

$$x(u) = A \cdot u^3 + B \cdot u^2 + C \cdot u + D \tag{6.2a}$$

$$\dot{x}(u) = \frac{dx}{du} = 3A \cdot u^2 + 2B \cdot u + C \tag{6.2b}$$

$$\ddot{x}(u) = \frac{d^2x}{du^2} = 6A \cdot u + 2B, \tag{6.2c}$$

where u represents a distance along the chord length between a pair of toolpath points. The following contains one solution to the set of spline parameters $\{A, B, C, D\}$.

Position Boundary Condition

The first set of equations comes from the assumption that for any pair of toolpath points, the first point coincides with the start of the spline and the second point coincides with the end of the spline. This creates the two equations,

$$\begin{aligned} [x(0)]_k &= D_{x_k} = X_k \\ [x(l_k)]_k &= A_{x_k} \cdot l_k^3 + B_{x_k} \cdot l_k^2 + C_{x_k} \cdot l_k + D_{x_k} = X_{k+1}. \end{aligned} \tag{6.3}$$

The k's in equation 6.3 represent which pair of toolpath points is being considered. By putting (6.3) into matrix form,

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \\ l_k^3 & l_k^2 & l_k^1 & 1 \end{bmatrix}}_{L_k^0} \underbrace{\begin{bmatrix} A_{x_k} \\ B_{x_k} \\ C_{x_k} \\ D_{x_k} \end{bmatrix}}_{\Theta_{x_k}} = \underbrace{\begin{bmatrix} X_k \\ X_{k+1} \end{bmatrix}}_{\zeta_{x_k}} \tag{6.4}$$

the position boundary conditions for all splines can be stacked, creating $2N$ equations, with $4N$ unknowns, where $N+1$ is the number of toolpath points.

$$\underbrace{\begin{bmatrix} L_1^0 & 0_{2\times4} & \cdots & 0 \\ 0_{2\times4} & L_2^0 & & \\ \vdots & & \ddots & \\ 0 & & & L_N^0 \end{bmatrix}}_{L^0} \underbrace{\begin{bmatrix} \Theta_{x_1} \\ \Theta_{x_2} \\ \vdots \\ \Theta_{x_N} \end{bmatrix}}_{\Theta_x} = \underbrace{\begin{bmatrix} \zeta_{x_1} \\ \zeta_{x_2} \\ \vdots \\ \zeta_{x_N} \end{bmatrix}}_{\zeta_x^0} \tag{6.5}$$

$$L^0 \cdot \Theta_x = \zeta_x^0 \quad \text{(2N Eqns.)} \tag{6.6}$$

Velocity and Acceleration Boundary Condition

The next set of equations is generated from the assumption that the 1st and 2nd order tangents at each toolpath point are the same for the two splines that intersect that point. This implies that the velocity and acceleration are the same where two spline segments meet. This forms a set of two equations and six unknowns.

$$\begin{aligned} [\dot{x}(l_k)]_k = [\dot{x}(0)]_{k+1} &\rightarrow 3A_{x_k} \cdot l_k^2 + 2B_{x_k} \cdot l_k + C_{x_k} = C_{xk+1} \\ [\ddot{x}(l_k)]_k = [\ddot{x}(0)]_{k+1} &\rightarrow 6A_{x_k} \cdot l_k + 2B_{x_k} = 2B_{x_{k+1}} \end{aligned} \tag{6.7}$$

These two equations create the following matrix form that contains the parameters for two splines.

$$\underbrace{\begin{bmatrix} 3l_k^2 & 2l_k & 1 & 0 & \vdots & 0 & 0 & -1 & 0 \\ 6l_k & 2 & 0 & 0 & \vdots & 0 & -2 & 0 & 0 \end{bmatrix}}_{L^1_{k,k+1}{}^2} \underbrace{\begin{bmatrix} A_{x_k} \\ B_{x_k} \\ C_{x_k} \\ D_{x_k} \\ A_{x_{k+1}} \\ B_{x_{k+1}} \\ C_{x_{k+1}} \\ D_{x_{k+1}} \end{bmatrix}}_{\begin{bmatrix} \Theta_{x_k} \\ \Theta_{x_{k+1}} \end{bmatrix}} = \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\zeta_{x_k}} \tag{6.8}$$

Equation 6.8 can be stacked to form a set of $2N-2$ equations that represent all of the spline equations.

$$\underbrace{\begin{bmatrix} L^{12}_{1,2_{(2\times 8)}} & 0_{(2\times 4)} & & \\ 0_{(2\times 4)} & L^{12}_{2,3_{(2\times 8)}} & & \\ & & \ddots & \\ & & & L^{12}_{N-1,N_{(2\times 8)}} \end{bmatrix}}_{L^0} \underbrace{\begin{bmatrix} \Theta_{x_1} \\ \Theta_{x_2} \\ \vdots \\ \Theta_{x_N} \end{bmatrix}}_{\Theta_x} = \underbrace{\begin{bmatrix} 0_{2(N-1)\times 1} \end{bmatrix}}_{\zeta_x^{12}} \tag{6.9}$$

$$L^{12} \cdot \Theta_x = \zeta_x^{12} \quad \text{(2N-2 Eqns.)} \tag{6.10}$$

Extra Boundary Condition

Considering the set of equations formed by equations 6.6 and 6.10, there are currently $4N-2$ sets of equations, with $4N$ unknowns. Two more equations must be found to simultaneously solve the parameters for all of the unique cubic splines. These two equations are created by specifying a constant value for the tangents, or velocities, at the start and end point of the toolpath. This will fill in the equations that were omitted when forming the velocity and acceleration boundary equations.

$$\begin{aligned} &[\dot{x}(0)]_1 = \dot{X}_1 \rightarrow C_{x_1} = \dot{X}_1 \\ &[\dot{x}(l_N)]_N = \dot{X}_{N+1} \rightarrow 3A_{x_N} \cdot l_N^2 + 2B_{x_N} \cdot l_N + C_{x_N} = \dot{X}_{N+1} \end{aligned} \tag{6.11}$$

Again, these equations can be put into matrix form and then stacked to create a set of equations for all of the spline parameters.

$$\underbrace{\begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}}_{L^{initial}} \underbrace{\begin{bmatrix}A_{x_1}\\B_{x_1}\\C_{x_1}\\D_{x_1}\end{bmatrix}}_{\Theta_{x_1}} = \underbrace{\begin{bmatrix}X_1^u\end{bmatrix}}_{\zeta_{initial}} \Bigg| \underbrace{\begin{bmatrix}3l_N^2 & 2l_N & 1 & 0\end{bmatrix}}_{L^{final}} \underbrace{\begin{bmatrix}A_{x_N}\\B_{x_N}\\C_{x_N}\\D_{x_N}\end{bmatrix}}_{\Theta_{x_N}} = \underbrace{\begin{bmatrix}X_{N+1}^u\end{bmatrix}}_{\zeta_{final}} \tag{6.12}$$

$$\underbrace{\begin{bmatrix} L^{initial} & \vdots & 0_{2\times 4(N-2)} & \vdots & 0_{1\times 4} \\ 0_{1\times 4} & & & & L^{final}\end{bmatrix}}_{L^{BC}} \underbrace{\begin{bmatrix}\Theta_{x_1}\\\Theta_{x_2}\\\vdots\\\Theta_{x_N}\end{bmatrix}}_{\Theta_x} = \underbrace{\begin{bmatrix}\zeta_{initial}\\\zeta_{final}\end{bmatrix}}_{\zeta_x^{BC}} \tag{6.13}$$

$$L^{BC} \cdot \Theta_x = \zeta_x^{BC} \quad (2 \text{ Eqns.}) \tag{6.14}$$

Adding equation 6.14 to 6.6 and 6.10, the system of equations now contains 4N equations with 4N unknowns. By grouping these equations into one matrix, the set of spline parameters can be solved with traditional matrix techniques.

$$\begin{bmatrix}L^0\\L^{12}\\L^{BC}\end{bmatrix} \Theta_x = \begin{bmatrix}\zeta^0\\\zeta^{12}\\\zeta^{BC}\end{bmatrix} \tag{6.15}$$

$$L \cdot \Theta_x = \zeta_x. \tag{6.16}$$

6.2.2 Total Arc Length

Now that the parameters for the cubic splines have been solved for all three axes, the total arc length of the toolpath can be calculated. Unlike the chord lengths, which are linear distances between toolpath points, arc lengths are an approximation of the length of the spline segment that connects two toolpath points. An arc length is calculated by approximating a spline segment with a large number of linear segments. By combining the arc length estimates for each axis, a three dimensional arc length for the spline that follows the contour of the model can be calculated. This value is required for constructing the smooth feed profile.

For the arc length calculation between two toolpath points, the corresponding chord length is divided into a discrete number of steps Δu. At each step along the chord length, the parameters of the corresponding spline are used to calculate a position value along the spline, using equation 6.2a. Figure 6.2 illustrates this concept.

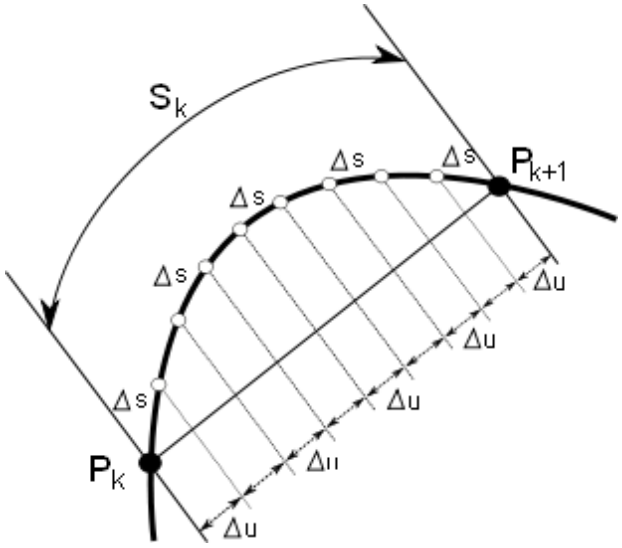


Figure 6.2: Arc Length Calculation

Linear distances are calculated between each of the position values found along the spline using

$$\Delta s_{k,j} = \sqrt{(x_{k,j} - x_{k,j-1})^2 + (y_{k,j} - y_{k,j-1})^2 + (z_{k,j} - z_{k,j-1})^2}. \tag{6.17}$$

Here, the 'k' index represents the spline segment, while the 'j' index represents the number of Δu's along the chord length. Summing the small linear distances, Δs creates the approximation for the arc length S. All of the arc lengths for each segment can be summed to create a total arc length for the entire toolpath.

6.2.3 Smooth Feed Profile

The total arc length of all the splines that pass through the toolpath points of the model is used to generate the smooth feed profile for the system. This profile can then be used to create a trajectory for all three of the motor axes. As mentioned, the profile is based on

continuous jerk motion, making the position profile a fifth-order curve. Figure 6.3 shows the shape the profile. The equations describing this profile were created based on the parabolic equation used to describe the desired continuous jerk motion and were integrated to obtain the position equations. This set of equations are listed in appendix A.

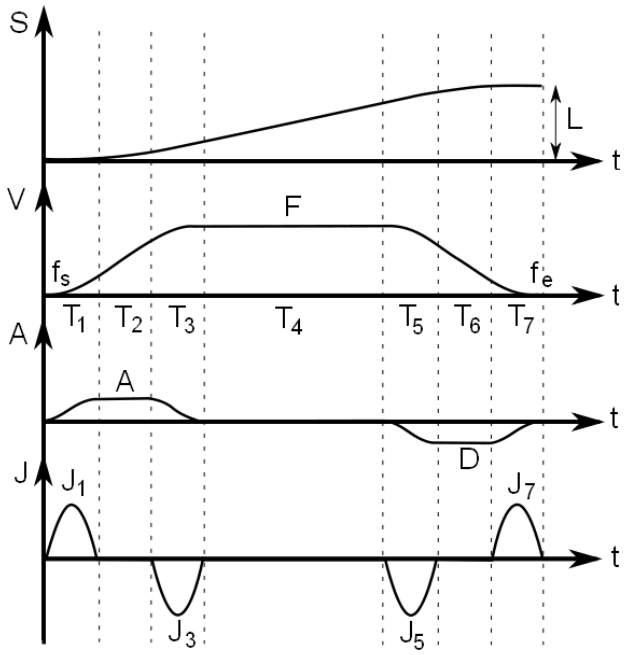


Figure 6.3: Smooth Feed Profile

The variables J, A, D and F used in the feed profile equations represent the jerk, acceleration, deceleration and feedrate values respectively. The sampling time T_s is also used in these calculations. These variables are selected prior to generating the smooth trajectory and must be selected based on the capabilities of the machining system. To ensure that the specified profile parameters will construct a proper profile, a number of conditions must be met.

Jerk Condition

To ensure that the jerk parameter does not contradict the maximum acceleration specification, the jerk J must adhere to,

$$J \leq J_{limit} = \frac{A_{max}}{T_s} \tag{6.18}$$

Otherwise the jerk will be set to $J = \frac{A_{max}}{T_s}$.

Acceleration Condition

To ensure that the increasing and decreasing acceleration periods, in segments T_1 and T_3, do not overlap, the following condition must be satisfied.

$$T_2 = \frac{F - f_s}{A} - \frac{T_1 + T_3}{2} \leq 0 \tag{6.19}$$

If the calculated time of T_2 is negative, then T_2 is set to zero and the acceleration must be recalculated using equation 6.20 to correct the feed profile. This case usually occurs when the acceleration is too low.

$$A = \sqrt{2J_1 \mid F - f_s \mid /3} \cdot sign(F - f_s) \tag{6.20}$$

Deceleration Condition

The same situation arises for deceleration in segments T_5 to T_7 as for acceleration. The deceleration must satisfy

$$T_6 = \frac{f_e - F}{D} - \frac{T_5 + T_7}{2} \leq 0 \tag{6.21}$$

If it does not, T_6 is set to zero and the deceleration term is recalculated according to

$$D = \sqrt{2J_5 \mid f_e - F \mid /3} \cdot sign(f_e - F). \tag{6.22}$$

This condition usually fails when the deceleration is too low, causing the T_6 segment of the profile to be overlapped by the increasing/decreasing deceleration.

Travel Length Condition

Finally, the length of the constructed profile is checked using a combination of all of the variables provided, as show in,

$$T_4 = \frac{1}{F}[L - (\alpha F^2 + \beta F + \gamma)] \leq 0. \tag{6.23}$$

where,

$$\alpha = \tfrac{1}{2A} - \tfrac{1}{2D}, \quad \beta = \tfrac{3}{4}(\tfrac{A}{J_1} + \tfrac{D}{J_5}), \quad \gamma = \tfrac{3}{4}(\tfrac{Af_s}{J_1} + \tfrac{Df_e}{J_5}) - \tfrac{f_s^2}{2A} + \tfrac{f_e^2}{2D}. \tag{6.24}$$

If the constant velocity duration T_4 is negative, then the motion is too short for the specified feedrate. This must be recalculated to a lower feedrate using,

$$F = \frac{-\beta + \sqrt{\beta^2 - 4\alpha(\gamma - L)}}{2\alpha}. \tag{6.25}$$

Motion Quantization

Once all of the conditions have been satisfied, the time duration of the profile must be checked to ensure that it is an integer multiple of the sampling period T_s. This will ensure that when the profile is run in a discrete system, the profile will be reconstructed properly and not approximated. This is ideal to maintain the carefully constructed smooth motions of the feed profile. If the profile is found to not be an integer multiple of the sampling period, an adjustment factor is calculated based on the total time $T_{\sum}$ and the adjusted total time $T'_{\sum}$.

$$\alpha_T = \frac{T'_{\sum}}{T_{\sum}} = \frac{N_{\sum} T_s}{T_{\sum}}, \text{ where } N_{\sum} = round\left(\frac{T_{\sum}}{T_s}\right). \tag{6.26}$$

The time segments of the feed profile are then adjusted based on the factor α_T as follows,

$$T'_i = \alpha_T \cdot T_i, \text{ for i = 1,...,7.} \tag{6.27}$$

Using the adjusted time segment values, a new feedrate must be solved that satisfies equation 6.28. This was found using the Newton-Ralphson method.

$$\alpha_0 \cdot F'^3 + \alpha_1 \cdot F'^2 + \alpha_2 \cdot F' + \alpha_3 = 0. \tag{6.28}$$

The new feedrate and segment times can then be used to calculate new values for the jerk, acceleration and deceleration parameters, using the following equations,

$$\begin{aligned} A' &= \tfrac{F'-f_s}{T'_1+T'_2} \quad J'_1 = \tfrac{3(F'-f_s)}{2T'_1(T'_1+T'_2)} \\ D' &= \tfrac{f_e-F'}{T'_5+T'_6} \quad J'_5 = \tfrac{3(f_e-F')}{2T'_5(T'_5+T'_6)}. \end{aligned} \tag{6.29}$$

Once the feed profile has been adjusted to fit with the sampling time of the system, the equations in appendix A can be used to produce the smooth profile.

6.2.4 Interpolate Smooth Toolpath

The algorithm uses a 1st order Taylor Series based interpolation approach to generate the intermediary points along the toolpath. Using the spline equations that have already been solved and the feed profile that has been constructed, new toolpath points are solved that will create the smooth, constant feed profile that follows the contours of the model surface. This new toolpath is generated for each axis and the spacing of the points will coincide with the sampling time of the system to create the desired smooth profile. The following equation is used to calculate u, the variable step size along a chord length between each new toolpath point.

$$u_i = u_{i-1} + \frac{\dot{s}_i \cdot T_s}{\sqrt{\dot{X}^2(u_{i-1}) + \dot{Y}^2(u_{i-1}) + \dot{Z}^2(u_{i-1})}}, \tag{6.30}$$

Equation 6.30 is developed using a 1st-order Taylor Series approximation. The $\dot{s}_i$ term is the feedrate constructed from the feed profile. The terms X_u, Y_u and Z_u are the velocity terms calculated from equation 6.2b, using the previous value of u.

Once u has been calculated, the corresponding spline of the form of equation 6.2a is used to calculate the new position points for each axis. This algorithm uses the interpolation method of equation 6.30, the feed profile constructed using the equations in section 6.2.3 and the spline parameters identified for equations 6.2 to create a smooth trajectory profile. When the individual trajectories of each axis are combined, the resultant trajectory will follow the form of figure 6.3. The position contour will be smooth, the feed profile will have little velocity fluctuations and the acceleration and jerk profiles will be continuous. This new toolpath or trajectory, will allow the 3-axis CNC lathe to machine faster with less feed rate fluctuations, while achieving a smooth surface finish.

6.3 Geometry Smoothing Algorithm

When cutting table legs on the 3-axis CNC lathe system, as described in chapter 3, the custom features often range anywhere from $1mm$ to $5mm$ in height. Sudden changes in height are difficult to cut, as the radial x-axis is required to move quickly to cover the distance in a short amount of time. The radial axis must move faster for height changes that are located on larger diameter cross-sections. This is because points farther away

from the centre of a rotating body spin faster. A typical table leg has a maximum radius of $40mm$. This makes the most difficult feature to machine a $5mm$ height change, from a radius of 35 to $40mm$.

When machining these features, the ASMC controller for the x-axis exhibits overshoot in terms of following the toolpath points. This error is due to the sudden changes in the surface of the table leg, requiring the controller and system to react quickly. The quick response brings in large acceleration values, which lead to the errors in tool positioning. The severity of the overshoot is proportionally related to speed and appears physically as gouging in the wood surface finish. Visible defects in a table leg are unacceptable because the aesthetic appeal of the final product is the primary reason for machining.

The algorithm proposed in this section attempts to reduce any errors in the surface finish created during machining by altering the CAD model of the part before machining occurs. This method will identify problem areas in a model's geometry caused primarily by sudden height changes, using simulations. The algorithm will change the position of individual toolpath points on the model's surface in these identified areas. This sacrifice in geometry will effectively smooth the sharp transitions, allowing the toolpath to be followed with less error.

By modifying a model's geometry in this fashion, a toolpath can be run at a faster feedrate without creating surface defects that make the finished work piece unusable. Once the algorithm has modified the geometry, these modifications must be approved by the operator before machining. Unlike defects, which mar the physical appearance of the finished product, smoothed edges still look visually acceptable. If the original shape of the model is altered beyond an acceptable amount, the operator will have to reduce the desired feedrate and run the algorithm again. This algorithm also removes the need to add smoothing to edges manually, which was a common practice when creating models for the 3-axis CNC lathe system. The algorithm will be able to smooth a given model by the minimum amount required to reach a desired feedrate. This will save time in both model design and machining.

The flow chart depicted in figure 6.4 demonstrates the logic behind the geometry modifying algorithm. Given a model that conforms to the geometry restrictions imposed by the 3-axis lathe system, the algorithm from section 6.2 is used to generate a trajectory at a desired feedrate. Using the model of the 3-axis lathe system developed in chapter 4, a

simulation of the output for the x-axis is generated. Only the x-axis needs to be simulated because it controls the depth of cut of the tool and is the axis prone to overshooting position. The simulated tracking error data can be analysed for problem geometry. Peaks can be identified which represent errors in the cutting tool positioning that are greater than a desired limit. These peaks, which directly correspond to specific toolpath points, can then be moved, along with neighbouring points, by a small defined increment. This will create the desired smoothing effect. Once a problem area has been smoothed, the algorithm must repeat the above process for the next problem area. After each iteration, the error peaks corresponding to the model regions being modified should become smaller and smaller. The iteration process will stop once all of the error peaks are below the desired limit. Using this method, each geometry region will be modified by the smallest increment to achieve the desired position tracking error performance. This implies that the model geometry will be modified only as much as necessary on a case by case basis for each identified problem region. Once the algorithm has finished its modifications the final model can viewed. If the sacrifices to geometry are acceptable, the model can be machined. This model will not have any surface defects greater than the set tolerance and can be machined at the desired feedrate.

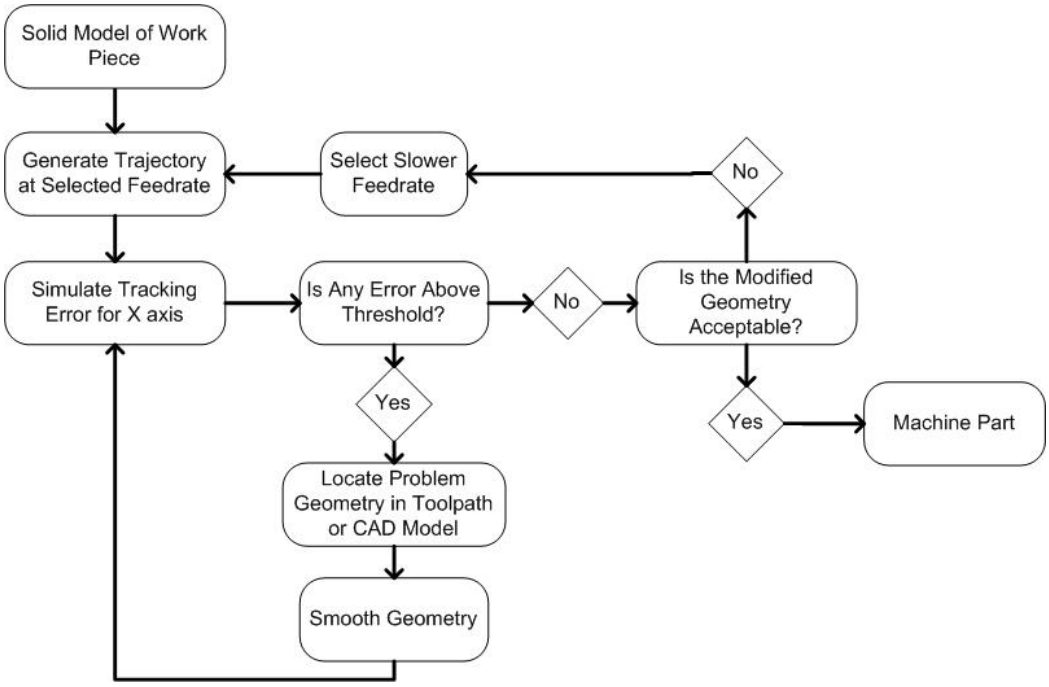


Figure 6.4: Algorithm Logic Diagram

Chapter 7

Test Cases

Two test cases have been developed to demonstrate the capabilities of the 3-axis CNC lathe and to evaluate the effectiveness of the changes to the system that have been described in the previous sections. This chapter presents the two test cases in detail, along with simulated and experimental results for machining these pieces. Each test model has its dimensions designed to be within the constraints of a typical table leg. The cross-section must be less than or equal to an $80mm$ diameter circle and the length of the part should be no more than $813mm$ in length. The first case consists of three basic shapes which are used to evaluate the speed and accuracy of the CNC lathe. The geometry presented attempts to highlight the unique machining ability of the lathe, while also demonstrating the speed and accuracy capabilities of the system in comparison to the original lathe design, as described in section 1.4. The second model is a simpler piece when considering geometry detail, but is difficult to machine nonetheless. This piece is used to demonstrate the concept of the smoothing algorithm proposed in section 6.3. The design of the piece is simpler to fully observe and evaluate the effects of the algorithm.

7.1 Test Case 1 - Speed Testing

7.1.1 Toolpath Feature Selection

The geometry of the three test pieces was selected to represent as many features from different table leg designs as possible. This was done to make the results from the ma-

chining experiments relevant for the intended physical application of the 3-axis CNC lathe. Although relevant for intended products, creating a standard test piece also establishes a basis for comparison. In the future, if a new toolpath method is developed or a new model and controller are implemented, these test pieces can be used as a comparison tool. The geometry used in this model will remain representative of the common features found in table leg designs, allowing new methods to test how they handle the different geometric features. Future modifications to the lathe system can use the results and observations drawn from these experiments as a gauge for their success.

Essentially, the creation of a table leg design is limited only by the imagination of the designer. This implies that the number of possible features that could be on a table leg is limitless. Already, from an existing set of fifty table leg designs, there is a large variety of features. Some of these features include; helices, letters, custom patterns and complex images. All of these features can be either embossed or debossed. Figure 7.1 illustrates some examples of these features. The toolpath features selected for the test case must capture elements from some, if not all, of the above mentioned features for the experimental results to be considered broad enough to apply to any table leg that could be machined.

(a) Pattern (b) Complex Shape (c) Letters

Figure 7.1: Table Leg Feature Examples

The first features to be selected are those for the shape of the bodies of the test pieces. The body of a table leg is the surface onto which all other features are added. The geometry

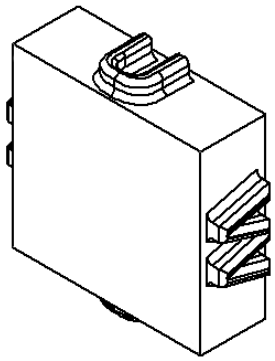
of a table leg body can be described as different polygons with a variable number of sides. The minimum number of sides is three, creating a triangular shape, and the maximum number of sides is infinity, representing a cylinder. The most common geometries in table leg design are the 4-sided and cylindrical shapes. For this reason they were selected as body shape features to be included in the test models. Including common body shapes makes comparison of the test pieces to full table legs more relevant. The cylindrical profile is the easiest table leg profile to machine. If only a cylinder was being machined, the tool would simply remain at a constant radius and the lathe could cut the body shape as fast as the rotary axis could turn. When adding additional features to the surface of this body profile, the difficulty in machining comes from the feature geometry alone. With a four sided polygon body profile, the machining is more difficult, in the sense that the tool axis must now accelerate and decelerate to track around the 90^o corners. When features are added to the surface of this shape, the demands from the cutting axis are increased even further. The more sides that are present in a table leg body, the easier it is to machine that body shape. Although not as common, the 3-sided table leg body profile was also selected for testing. The angles at which the corners of the polygon meet are 60^o, presenting an even greater challenge than that of the 4-sided body shape. The triangular body is considered to be a 'worst case scenario' for a table leg body due to the extreme geometry, and is chosen to see what adding additional features to the body shape will do to the performance of the 3-axis CNC lathe system. Selecting table leg body geometries as toolpath features is significant because, as mentioned, they are the base shape onto which all other features are added. Adding features to the surface of the table leg slows down the machining time. The results of this test case can hopefully be used as a base line for machining speeds of different table leg designs on the new 3-axis CNC lathe system.

The second toolpath feature to be selected needs to be from the set of features that encompasses all of the possible features that could be added to the surface of a table leg body. To choose appropriately, a worst case scenario feature was selected based on the constraints of the physical system. The maximum height of the feature was selected to be $5mm$. This height was chosen to make a suitable comparison to the features that were machined on the original lathe system. Typically, $5mm$ was the maximum height used for features on the original lathe. Any feature designed using this height parameter for the test case should be more difficult to machine than similar features with a lesser height. The tool axis must cover a greater distance in the same amount of time for features of larger height,

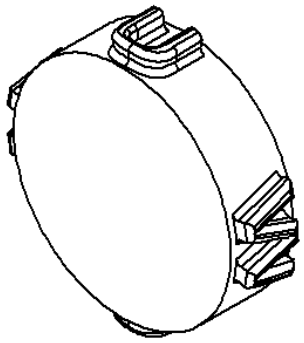
requiring greater speed and accuracy. Therefore, selecting a worst-case feature that uses this maximum height for the test case model should effectively represent all other features. Although the selected feature may be visually different from other features, if it is one of the most difficult to machine, then the test case results of this feature should also apply to other features.

The feature selected for the test case was letters. Letters are abrupt height changes on the body of a table leg. In control theory terminology, they come close to representing steps in height change, which are one of the most difficult features for a controller to accurately track. The letter feature cannot be a true step in height however, as it would be near impossible to machine without some error, which would cause gouging of the wood surface. Therefore the letter features must have at least a small slope during the height transition. Letters can easily be made to conform to the $5mm$ height restriction and many letters consist of multiple steps. This allows the testing of not just one sharp height change, but multiple repetitive height changes. The letter feature can also be made as both embossed and debossed. Although other features may appear more complicated, such as complex patterns or images, they are in fact composed of simpler features, such as smooth changes in height. Letters contain the most difficult feature to machine and are a suitable choice for representing all other features as a worst-case scenario. Observations made while machining letters should be applicable to most other features. The limits in terms of speed and accuracy observed while machining these test pieces should therefore apply to regular table leg designs as well.

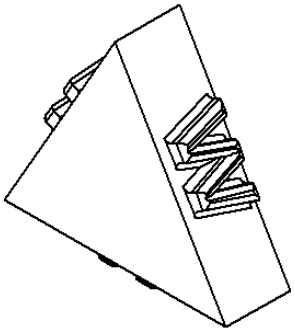
The three test pieces that were designed are displayed in figure 7.2 below. As can be seen in the models, all of the features discussed above are present in the designs. There are three test case segments with different body shape profiles, each $20mm$ thick. Letter features have been placed on the surface of all three pieces. These features have been embossed and conform to the $5mm$ height constraint. The letters chosen for the test pieces are 'U' and 'W'. These letters were selected as features because they have multiple vertical line segments that make up their structure. This allows testing of multiple steps up and down with one letter. Also, corners, like in the 'W', are difficult features to machine due to the tight constraints on geometry. The letter 'U' also adds the variable of having a curved edge. The size of the cross section of the test piece bodies and features are designed to fit within the $80mm$ diameter restriction imposed on table leg designs.



(a) Square Body



(b) Circular Body



(c) Triangular Body

Figure 7.2: Test Case 1 Models

7.1.2 Machining Results

The results of each test piece are analyzed separately. The trajectory for each segment was tested prior to machining to determine the maximum feedrate that can be reached while maintaining a position following error of $0.5mm$ on the model surface. This position following error was chosen because it was found that this much error in a wood material does not show up clearly in the finished surface. The intended table leg products have a large aesthetic component, which makes having a surface finish that looks good an important requirement. The results from the test models are also compared to the approximate results from the previous 3-axis CNC lathe design. Unfortunately, the results of the different machining tasks that the previous lathe design was used for were not recorded. Encoder data and similar measurements were not stored, but only used as feedback during the machining process. Also, the lathe was not used to machine any baseline models for setting testing standards, but used to machine table legs. However, based on operator knowledge, it has been established that for extreme surface geometry, similar to the geometry proposed for the test models, the maximum speed achieved by the 3-axis lathe was between 5 to $10rpm$, measured at the rotary axis. For the sake of comparison, it will be assumed that the average machining speed of the rotating axis is $8rpm$ for the original lathe system. Speed was measured at the rotary axis for the previous lathe system to make a comparison to regular lathe machines, which typically measure machining speeds in rpm's of the rotating axis. The speed of the rotary axis was also how the trajectory generator was controlled for the old system. If the speed of the rotary axis was increased or decreased, the speeds of the other two axes were also altered by the same proportion. For this reason, the results for the speed of the rotary axis for the current lathe design will be presented for each of the test pieces to make a comparison with the previous lathe system.

For the four-sided polygon segment, the machining results are presented in figure 7.3. This three dimensional figure shows a visualization of the machining process. The blue lines represent the physical location of the tool as it followed the helical toolpath. This figure shows that the experimental cutting results appear to follow the intended toolpath accurately. Figures 7.4 to 7.6 show the experimental tracking error of each axis. As seen in 7.4, the error for the x-axis is below $0.5mm$, as desired. The remaining two axes also have low error results, with the rotational axis staying below 1^o of error. This confirms that the four sided polygon section is being machined within acceptable tolerances. Although

difficult to make out due to the amount of data that is being shown, there is a great deal of oscillation present in the figures. The peaks in the error that are the source of this oscillating motion are caused by the letter features on the model. The feedrate at which these results were achieved was approximately $83mm/s$. This corresponds to an average velocity of $18.16rpm$ at the rotary axis, as shown in figure 7.7. This corresponds to a 55.94% reduction in machining time compared to the previous 3-axis lathe system.

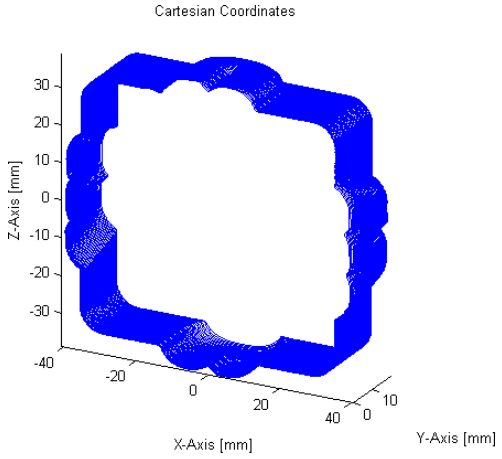


Figure 7.3: Rectangular Profile Experimental Results - 3D Output

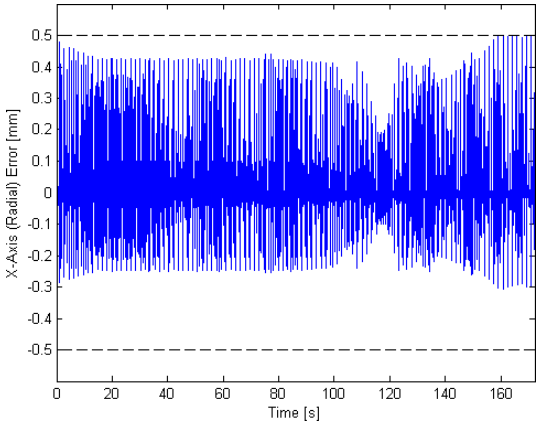


Figure 7.4: Rectangular Profile Experimental Results - Rotational Axis Error

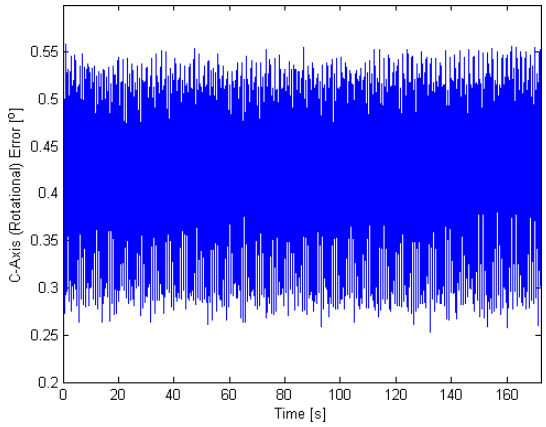


Figure 7.5: Rectangular Profile Experimental Results - Radial Axis Error

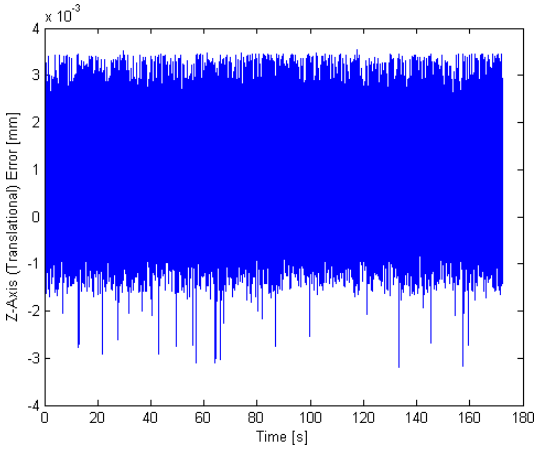


Figure 7.6: Rectangular Profile Experimental Results - Translational Axis Error

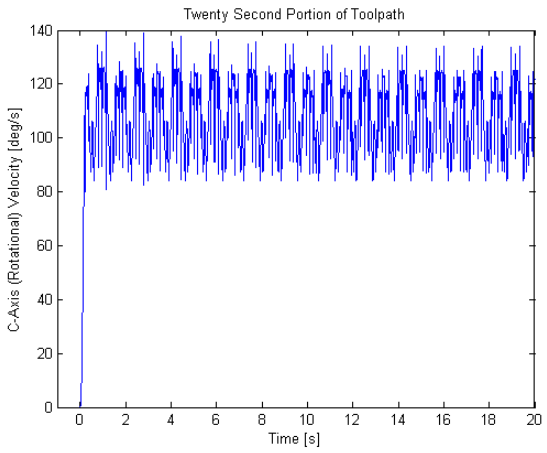


Figure 7.7: Rectangular Profile Experimental Results - C Axis Velocity Profile

The circular body test piece produced similar results to the four sided polygon segment. Figure 7.8 shows the three dimensional view of the results. Again, the system is able to follow the toolpath accurately. Figures 7.9 to 7.11 show the error plots for each axis. These plots are all within acceptable ranges, confirming that the model was machined accurately. The feedrate at which the error threshold is reached for this test piece is approximately $79mm/s$. This corresponds to an average speed of $16.99rpm$ at the rotary axis and a 52.91% machining time reduction over the former 3-axis lathe system. Although only slightly slower than the feedrate obtained for the square profile test piece, the circular profile is attributed to causing this difference. When machining a plain body, the cylinder should be the fastest; however, the addition of features has a greater impact on it than for other shapes. The change in height for each feature is presented as a nearly instantaneous change for the tool axis, which is difficult to machine quickly. In the 4-sided test case, the height changes of the features are not perpendicular to the tool axis. This gives the features a less steep slope for the tool axis to follow, which should be easier to machine and explain the difference in feedrates.

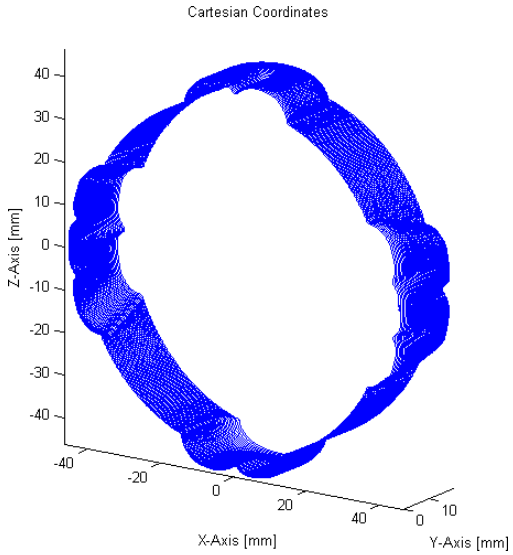


Figure 7.8: Circular Profile Experimental Results - 3D Output

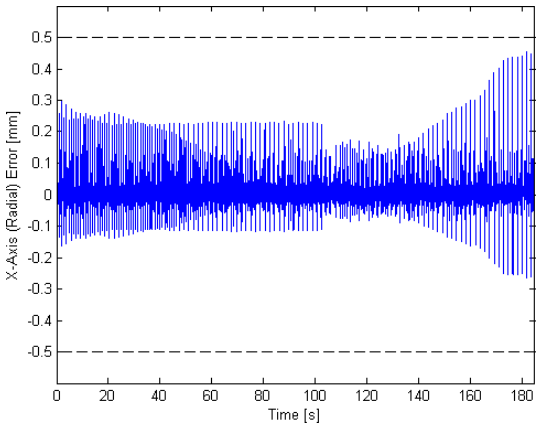


Figure 7.9: Circular Profile Experimental Results - Rotational Axis Error

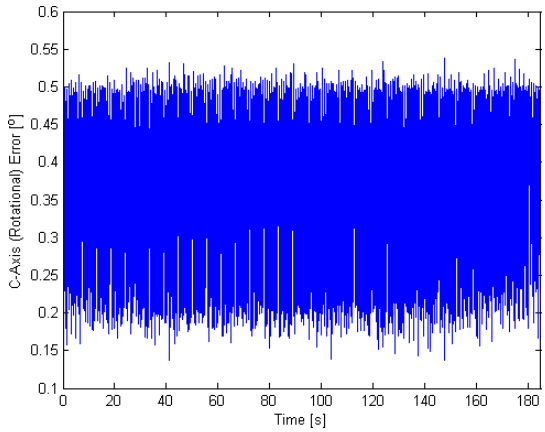


Figure 7.10: Circular Profile Experimental Results - Radial Axis Error

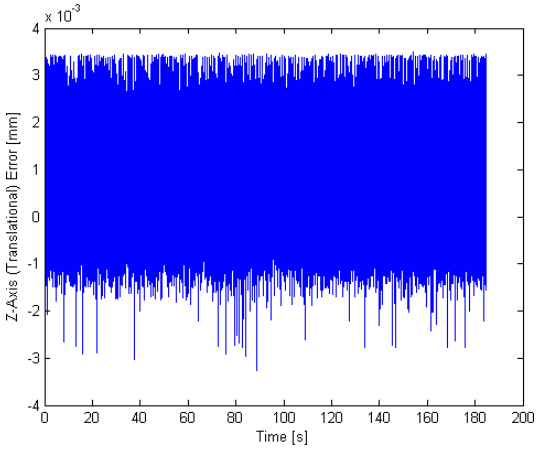


Figure 7.11: Circular Profile Experimental Results - Translational Axis Error

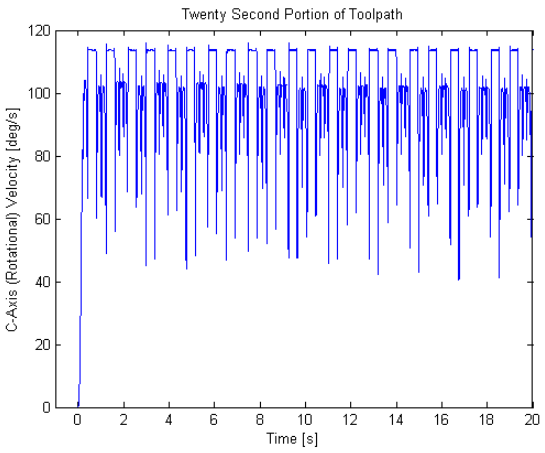


Figure 7.12: Circular Profile Experimental Results - C Axis Velocity Profile

Figure 7.13 displays the visual results of the three sided polygon body segment. Following the same trend as the other two segments, this segment appears to be machined accurately. The error plots shown in figures 7.14 to 7.16 confirm that the machining tolerance of $0.5mm$ has been met. The feedrate that achieved these results was approximately $76mm/s$. This corresponds to a speed of $19.08rpm$ at the rotary axis and an apparent 58.07% reduction in machining time compared to the speed of the previous 3-axis lathe system. These results are slightly misleading, as the average feedrate of the triangular test piece is the slowest of the three, but the speed of the rotational axis is the highest. These results are explainable because when the tool is traversing around a corner of the work piece, the rotational axis must turn the part quickly to maintain the desired feedrate, since the slope of the work piece with respect to the tool becomes steep. As has been mentioned, all of the axes of the lathe are programmed to work together during the machining. This speed increase essentially raises the average speed of the rotational axis, while the overall feedrate for the test case is still slowed due to the features and the fast changing slope of the triangle.

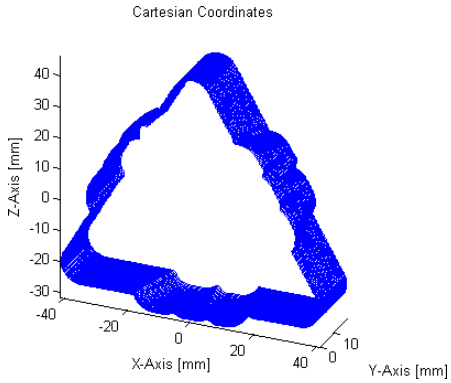


Figure 7.13: Triangular Profile Experimental Results - 3D Output

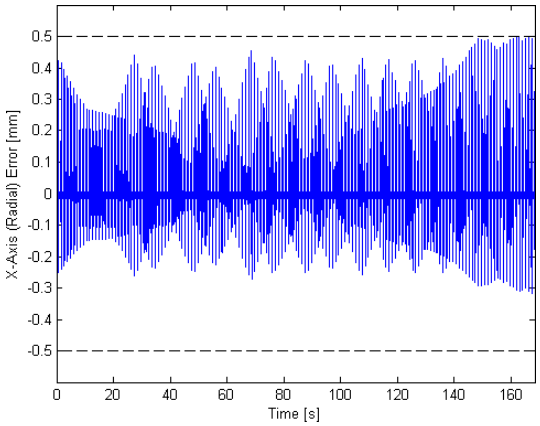


Figure 7.14: Triangular Profile Experimental Results - Rotational Axis Error

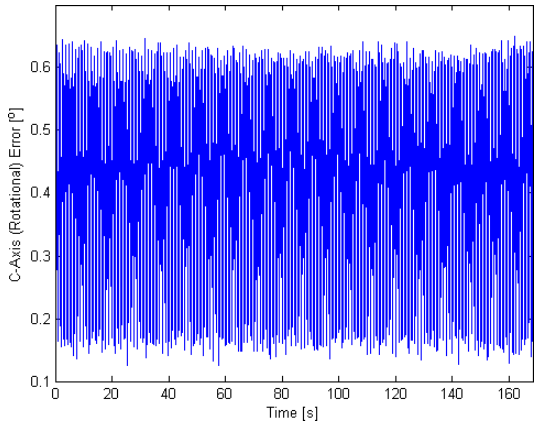


Figure 7.15: Triangular Profile Experimental Results - Radial Axis Error

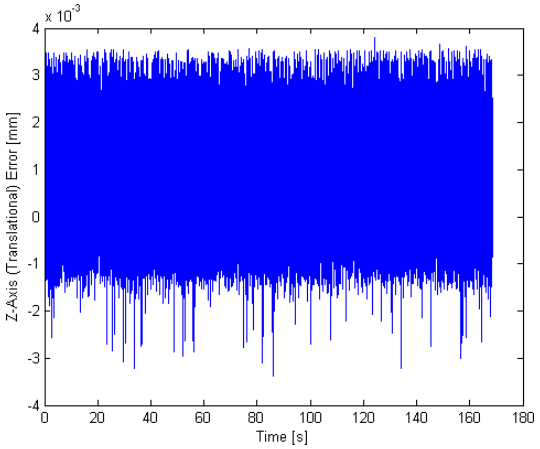


Figure 7.16: Triangular Profile Experimental Results - Translational Axis Error

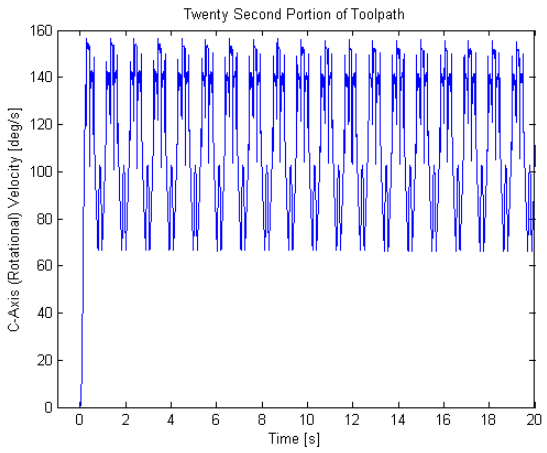


Figure 7.17: Triangular Profile Experimental Results - C Axis Velocity Profile

The experimental results presented above indicate that the 3-axis lathe system is performing well. The system is able to consistently achieve the desired tracking error performance, which allows the test cases to meet aesthetic requirements by avoiding gouging. The minimum machining feedrate of the test cases was $76mm/s$. This was the speed of the three-sided polygon segment. Although all three of the cases had similar features on the surface, the shapes of the bodies also affected the difficulty of the machining. The triangular body shape presents features with fast changing slopes to the tool axis, which my have affected the feedrate of the system. Higher acceleration requirements for the cutting axis make it harder to control, and could explain why the speed for machining the three sided polygon is lower than the other two segments while achieving the same accuracy. However this feedrate is only $7mm/s$ slower than the fastest of the cases.

The minimum speed out of the test cases for the rotational axis was found to be approximately $16.99rpm$. This machining speed reduces the machining time by approximately 52.91% when compared to the an the average machining speed of $8rpm$ observed during use of the original 3-axis CNC lathe system. This result highlights the effectiveness of the improvements to the design of the 3-axis CNC lathe system. The test results show that when machining the worst case scenario body profiles and worst-case-scenario toolpath features, the lathe system is able to machine faster than the previous system. The improvements made to the 3-axis CNC lathe system allow it to machine the most difficult features at least two times faster than the previous system.

These improvements to machining feedrate were made possible by the implementation

of all the theory described in chapters 4 to 6. Switching from a servo motor to a stepper motor allowed for larger torque demands and faster response times. The development of a model for the speed limiting x-axis allowed, among other things, the implementation of a complicated yet rigorous control law to improve tracking error. Applying a trajectory generator that not only created smooth trajectories for each axis, but also planned the movements of each axis to work together improved the overall feedrate of the system.

These positive results indicate the improvements made to the 3-axis CNC lathe system are a step in the right direction. They also imply that the results from this test model can be used as a performance baseline. In the future, if further modifications are made to the system, the accuracy and feedrate results presented above can be used as a baseline to gauge the effectiveness of said modifications. Also, these results, as they are specific to three common table leg body shapes and contain the worst case scenario features for each body shape, can be used for selecting machining feedrates during production. Any table leg that is similar to one of the three body profiles presented here will have a similar maximum feedrate limit in order to achieve a surface finish accuracy of $0.5mm$. If a table leg does not have surface features with the same extremes as the ones in the test model, the lathe system will most likely be able to machine it even faster. This will help with selecting a feedrate for the trajectory generator that will achieve the desired results during machining.

7.2 Test Case 2 - Smoothing Algorithm

The algorithm described in section 6.3 for smoothing the geometry of a work piece was tested on a simple model. Shown in figure 7.18, the model consists of a cylindrical cross section with a maximum diameter of $80mm$ and a constant thickness of $5mm$. Around the perimeter of the cross section are eight equally spaced height changes. The height changes begin at $3mm$ and increase by $1mm$ increments up to $10mm$. Steps of different height were chosen as features for the test case of the algorithm because, as mentioned for the previous test case, steps are one of the hardest features to machine quickly and accurately. Smoothing steps in the cutting axis motion, which are viewed as edges on the actual work piece, should effectively demonstrate any benefit being gained by the proposed algorithm. Unlike in the previous test case, the maximum step height is $10mm$, not $5mm$. The $10mm$

step height was chosen to further illustrate the effects of the smoothing algorithm, as a larger height change should generate a greater amount of smoothing.

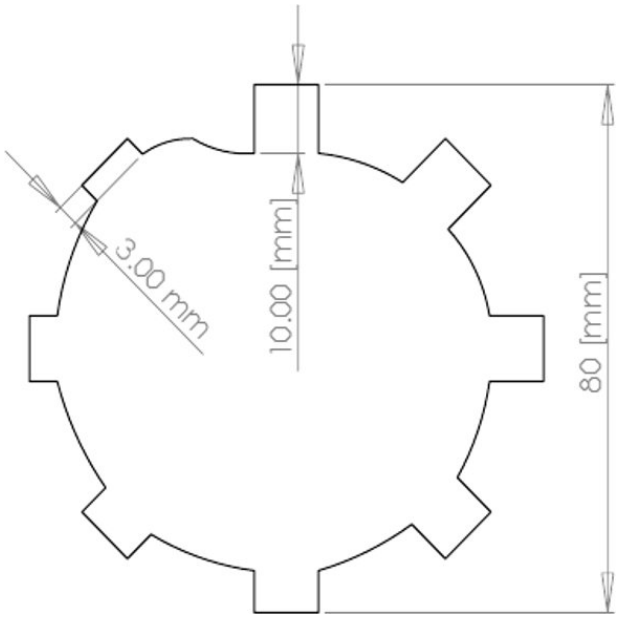


Figure 7.18: Case 2 - Test Piece Before Modifications

To begin the test case, a maximum position tracking error of $0.5mm$ was selected. This maximum was chosen because surface defects of this magnitude are not easily noticed in wood machining. The maximum can be altered however for different applications and preferences. Using the toolpath for the test piece and the x-axis model developed in chapter 4; Matlab/Simulink was used to simulate the system output until a maximum trajectory feedrate was found that satisfied the minimum error criterion. This feedrate was $70mm/s$. The model was then machined on the physical 3-axis lathe system at a feedrate of $70mm/s$ for experimental validation. The plot in figure 7.19 shows a portion of the tracking error results for the x-axis for machining the test piece at the specified feedrate. The data selected for display is the data for a single revolution of the test piece. After the first revolution, the remaining revolutions should all be identical, as the step features do not change. As seen in the figure, all of the tracking error is below $0.5mm$. The boundary is represented by the dashed black line located at $\pm 0.5mm$ on figure 7.19. By comparing the error plot to the toolpath, the peaks in the error plot are confirmed to be occurring at the edges of the steps, with a number of smaller peaks present after the peak. The large peak

is due mainly to overshoot in the controller of the x-axis and the smaller peaks are known as the 'ringing' effect, which occurs after an overshoot [9].

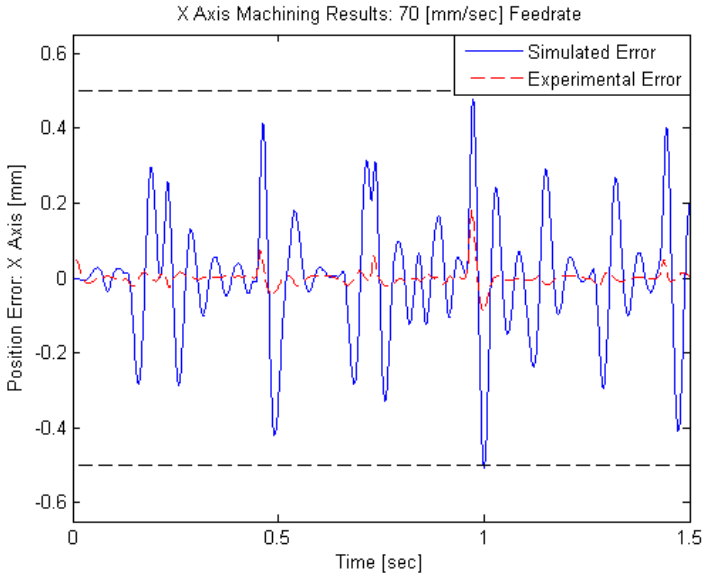


Figure 7.19: Test Results for 70 [mm/sec] Feedrate - Original Geometry

A new feedrate of $100mm/s$ is selected as the increased target machining feedrate for this test case. The work piece was machined again on the 3-axis lathe at the higher feedrate to obtain the experimental results. Figure 7.20 shows a portion of the simulated and experimental tracking error results for the x-axis at this increased speed. As before, the figure shows the data for a single revolution of the rotary axis. As indicated in the plot, there are error peaks in both the simulated and experimental data which exceed the set limit of $0.5mm$. Using the toolpath for comparison, each tracking error violation can easily be identified and associated with a region of geometry on the model. The error peaks identified at a $100mm/s$ feedrate are the same peaks as those identified at the $70mm/s$ feedrate test. The peaks have grown larger however due to the increase in speed. The overshoot caused by the controller gets worse when trying to navigate the sharp height transitions at a faster feedrate.

Using the methodology for the proposed geometry smoothing algorithm outlined in section 6.3, the fillet tool in SolidWorks is used as the smoothing mechanism for the proof-of-concept of the algorithm. Fillets are first added with a radius of $0.5mm$ to each edge

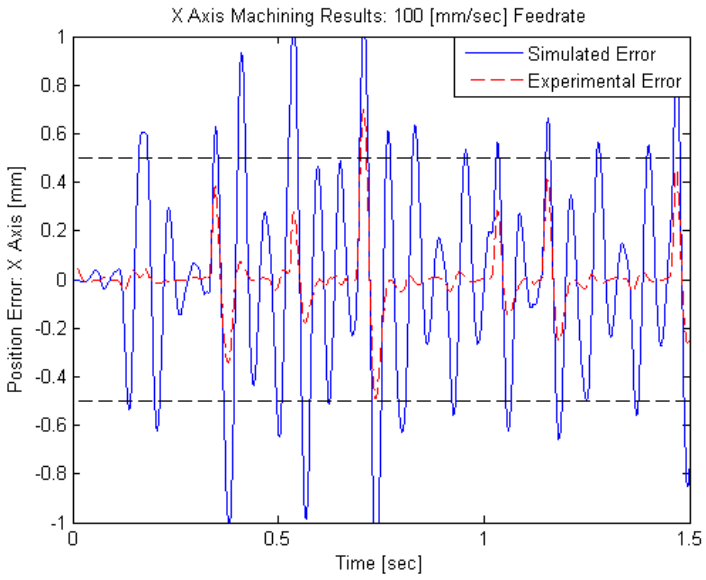


Figure 7.20: Test Results for 100 [mm/sec] Feedrate - Original Geometry

that has been identified to cause a tracking error in violation of the $0.5mm$ criterion. If the tracking error is not satisfied in simulation after this modification, the fillet radius is increased by $0.5mm$ for each edge that still fails. As indicated in the algorithm flow chart on page 81, this iterative process continues until the tracking error requirement is met for the entire model in simulation. The model with modified geometry was machined on the 3-axis lathe system to obtain experimental results. Figure 7.21 presents both the simulated and experimental error tracking plots for the modified test model. This figure only displays the x-axis data for the time required for the rotary axis to complete a single revolution. Figure 7.22 shows the resulting CAD model with modified geometry.

The results presented in figure 7.21, show that the 3-axis CNC lathe system is able to machine the test piece at the target feedrate of $100mm/s$. It does so while meeting the criterion of $0.5mm$ position tracking error for the x-axis. As can be observed in the model in figure 7.22, each edge that caused a violation in error tracking has been smoothed, to varying degrees depending on the severity of the violation. The rising and falling edges of each step are not smoothed identically in each case however. The $10mm$ step highlights this fact. Figure 7.22 shows that one of the bottom edges of the $10mm$ step has a radius of $2.5mm$, while the other bottom edge has a larger radius of $4.5mm$. This effect is

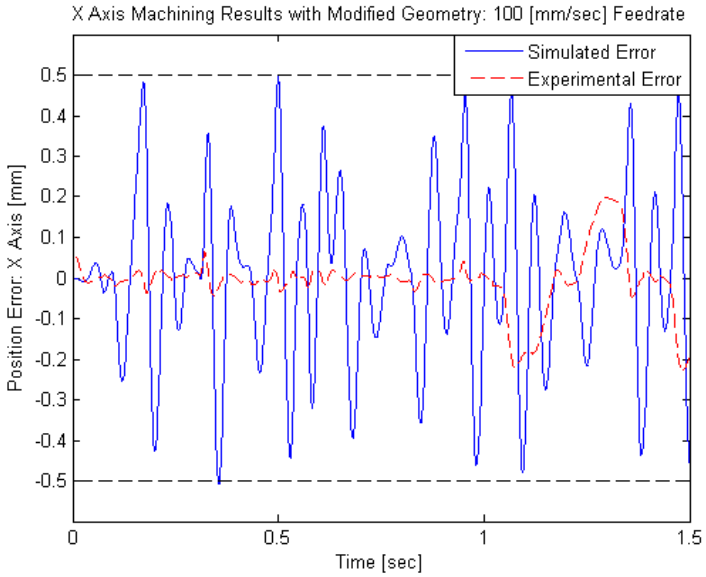


Figure 7.21: Test Results for 100 [mm/sec] Feedrate Modified Geometry

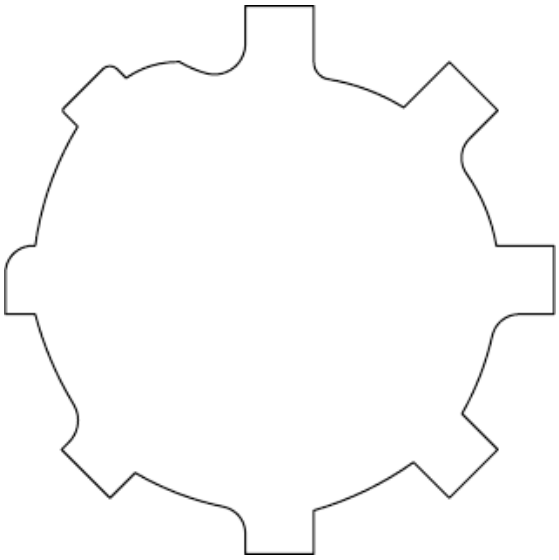


Figure 7.22: Case 2 - Test Piece After Modifications

created because the tool axis can move with precision in one direction, but not the other. This situation can arise due to inaccuracies in the system model, specifically the friction model. This effect may also be attributed to unmodelled dynamics present in the motor or ballscrew.

The system has experienced a 42.8% feedrate increase, while avoiding any unwanted surface defects, such as gouging. This has been accomplished using the proposed algorithm in section 6.3 for smoothing the geometry of the test model. As seen in figure 7.22, the modified model is quite a bit smoother than the abrupt transitions of the original model. The operator of the 3-axis lathe must qualitatively decide whether the sacrifices to geometry are acceptable. If this algorithm were to be used on a table leg model, the smoothing of some of the features on the leg could achieve a substantial reduction in machining time due to the length of the toolpath. This method is more plausible for products such as table legs, because they are primarily based on their aesthetic appearance and not dimensional tolerances. These results demonstrate the merit of this approach for creating an automated algorithm to accomplish the same purpose.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This work presented several modifications and additions to a 3-axis CNC lathe with the intent to improve the overall machining speed of the system. A servo motor replaced a stepper motor on the speed limiting axis of the system. The radial axis of the lathe was found to be the limiting axis of the system because the dynamic demands on the axis were much higher than those of any other. A 1st-order linear model was identified to describe the modified axis and adaptive sliding mode theory was implemented for control. In initial testing, the model appeared to describe the dynamics of the x-axis satisfactorily and the controller was able to provide accurate positioning. The new trajectory generator that was implemented was able to create a smooth trajectory for each axis of motion. By using a 5th-order polynomial to create a smooth feed profile with continuous jerk motion, each axis had continuous acceleration and velocity profiles as well. Smooth profiles allow a motor to more easily follow the desired commands.

From the discussions of the experimental test cases presented in Chapter 7, these modifications appear to be successful in improving the speed of the system. Three test pieces were machined, each representing a basic body shape that is used in table leg designs. These were circle, square and triangle cross-sections. The difficulty of machining these basic shapes was increased by adding letters to the surfaces. The letters were used to represent the many different types of features that can exist on a table leg surface. The

results of these experiments showed a successful increase in the machining speed of the CNC lathe compared to that of the original CNC lathe that was used before this work began. Each test case showed improvements to the reported speed from the original CNC lathe. The circular profile reduced machining time by approximately 52.91%, the square profile by 55.94% and the triangular profile by 58.07%, when comparing speeds of the rotational c-axis in rpm's. These results successfully demonstrated that the modifications to the 3-axis CNC lathe were able to increase the overall machining speed of the system while maintaining its ability to accurately machine complex sculptured surfaces. The results will also be of use as a new base line for speed testing if further modifications are made to the lathe system.

This work also looked at increasing the machining speed of the lathe system by modifying the geometry of a work piece. An algorithm was developed and presented in Section 6.3 which used the modelling knowledge of the system to simulate the machining output. By modifying a work piece's geometry, specifically by adding fillets to edges, the geometry could be smoothed, which allowed the generation of faster, yet accurate, trajectories. The effects of the geometry modifications could then be simulated to evaluate the improvements to the machining process.

The geometry sacrificing algorithm was also tested experimentally. A test case was created that consisted of several steps in height. By running the geometry smoothing algorithm on this test case, the algorithm could be evaluated by how much geometry was sacrificed to achieve a desired speed. The speed increase was found to be 42.8%, but the amount of geometry sacrificed is more of a qualitative measurement. In the test case, the smoothing did not appear to be overly excessive to achieve the desired feedrate of $100mm/s$ and still looked acceptable. The test case showed a positive result, indicating that the sacrifice of geometry in a model is a viable way of achieving machining speed gains.

Overall, the work presented in this thesis has been successful in improving the machining speeds of the 3-axis CNC lathe. The modifications to the lathe architecture have allowed for an approximate 52.91% reduction in machining time compared to that of the original CNC lathe. The algorithm used for sacrificing model geometry has been shown to effectively increase the speed of the system while maintaining a desired surface finish quality.

8.2 Future Work

Although the machining speed of the 3-axis CNC lathe system has been successfully increased, there are still many improvements that can be made to the system. For instance, currently the data transfer of the toolpath information from the PC to the FPGA board happens at one time. This takes a relatively long time, several minutes, to transfer this information and the amount of information that can be transferred is limited to the on-board memory of the FPGA board. A limit on the size of the toolpath dictates how large of a work piece can be machined. Currently, for a part with a maximum diameter of $80mm$, it can only be approximately $50mm$ in length. This drawback can be solved by implementing a data transfer scheme that moves only a small portion of the entire toolpath over to the FPGA board at one time. These small data transfers can continue while the work piece is being machined, which will allow that part to be as large as the physical limits of the system. Also, as mentioned in Section 6.2 the toolpath trajectory algorithm requires a large amount of physical memory to run. If more efficient methods for solving splines are investigated, the trajectory could be solved all at once, which would remove some of the feedrate fluctuation problems that developed when solving the trajectory in small pieces.

The 3-axis lathe is still currently in a prototype state. Consequently there are still many improvements that can be made to the user interface and mechanical system to polish the design. Additionally, there is still discussion on expanding the physical capabilities of the lathe by adding a 4th or even a 5th axis. The benefit of adding more sensors, such as accelerometers for improved control, is also being considered.

The model used to describe the x-axis of the system could be improved. As can be seen in the lathe simulation figures in section 7.2, the system output generated by the model is larger than the experimental data by approximately 1/3. By either conducting a re-identification of the current model or creating a new model that is perhaps more complex, the performance of the system can be improved. The ASMC performance would be improved by having more accurately identified plant parameters, allowing the system to improve its position tracking performance at higher speeds. The geometry sacrificing algorithm also uses the x-axis model in its calculations. A more accurate model could improve the geometry to speed ratios that are determined by the algorithm.

There is also a great deal of work to be done with the geometry sacrificing algorithm itself. Right now, the fillet feature in SolidWorks is used to create the geometry modifi-

cations to the work piece model. However, other methods of smoothing model geometry should be explored. A fillet uses a constant radius, whereas another type of curve, such as a custom spline, could be created that smooths model geometry enough to allow the machining speed of the system to increase, but does not remove as much material. Such a smoothing method could also be applied to the toolpath points itself, which would avoid the current step of modifying the model geometry in SolidWorks, which is time consuming.

It is also recommended that a relationship between model geometry and system dynamics be explored. Although this relationship would most likely be unique to a specific system, it would improve the speed of the geometry sacrificing algorithm. If the relationship between how specific geometry affected the velocity and accelerations of the system was known, then geometry could be modified to avoid situations that caused violations in the dynamic limits of the system; without the need for simulation of the model output. This would remove a time consuming portion of the algorithm, where it currently must use Matlab/Simulink to simulate the output of the x-axis. This relationship would allow the algorithm to scan the model geometry and modify the toolpath, avoiding the need to iteratively simulate the system output and regenerate the work piece toolpaths and trajectories.

APPENDICES

Appendix A

Feed Profile

A.1 Feed Profile Equations

Position:

$$s(\tau) = \begin{cases} \frac{-A}{10T_1^3} + \frac{A}{4T_1^2}\tau_1^4 + f_s\tau_1 & & 0 \le t < t_1 \\ s_{1e} + \frac{A}{2}\tau_2^2 + f_{1e}\tau_2, & s_{1e} = \frac{3}{20}AT_1^2 + f_sT_1, & t_1 \le t < t_2 \\ s_{2e} + \frac{A}{10T_3^3}\tau_3^5 - \frac{A}{4\tau_3^4} + \frac{A\tau_3^2}{2} + f_{2e}\tau_3, & s_{2e} = s_{1e} + \frac{AT_2^2}{2} + f_{1e}T_2, & t_2 \le t < t_3 \\ s_{3e} + f_{3e}\tau_4, & s_{3e} = s_{2e} + \frac{7AT_3^2}{20} + f_{2e}T_3, & t_3 \le t < t_4 \\ s_{4e} - \frac{D}{10T_5^3}\tau_5^5 + \frac{D}{4T_5^2}\tau_5^4 + f_{4e}\tau_5, & s_{4e} = s_{3e} + f_{3e}T_4, & t_4 \le t < t_5 \\ s_{5e} + \frac{D\tau_6^2}{2} + f_{5e}\tau_6, & s_{5e} = s_{4e} + \frac{3DT_5^2}{20} + f_{4e}T_5, & t_5 \le t < t_6 \\ s_{6e} + \frac{D\tau_7^5}{10T_7^3} - \frac{D\tau_7^4}{4T_7^2} + \frac{D\tau_7^2}{2} + f_{6e}\tau_7, & s_{6e} = s_{5e} + \frac{DT_6^2}{2} + f_{5e}T_6, & t_6 \le t < t_7 \\ & s_{7e} = s_{6e} + \frac{7DT_7^2}{20} + f_{6e}T_7 & \end{cases} \tag{A.1}$$

Velocity:

$$\dot{s}(\tau) = \begin{cases} f_s - \frac{A\tau_1^4}{2T_1^3} + \frac{A\tau_1^3}{T_1^2}, & & 0 \le t < t_1 \\ f_{1e} + A\tau_2, & f_{1e} = f_s - \frac{AT_1}{2}, & t_1 \le t < t_2 \\ f_{2e} + \frac{A\tau_3^4}{2T_3^3} - \frac{A\tau_3^3}{T_3^2} + A\tau_3, & f_{2e} = f_{1e} + AT_2, & t_2 \le t < t_3 \\ f_{3e}, & f_{3e} = f_{2e} + \frac{AT_3}{2}, & t_3 \le t < t_4 \\ f_{4e} - \frac{D\tau_5^4}{2T_5^3} + \frac{D\tau_5^3}{T_5^2}, & f_{4e} = f_{3e}, & t_4 \le t < t_5 \\ f_{5e} + D\tau_6, & f_{5e} = f_{4e} + \frac{DT_5}{2}, & t_5 \le t < t_6 \\ f_{6e} + \frac{D\tau_7^4}{2T_7^3} - \frac{D\tau_7^3}{T_7^2} + D\tau_7, & f_{6e} = f_{5e} + DT_6, & t_6 \le t < t_7 \\ & f_{7e} = f_{6e} + \frac{DT_7}{2} & \end{cases} \tag{A.2}$$

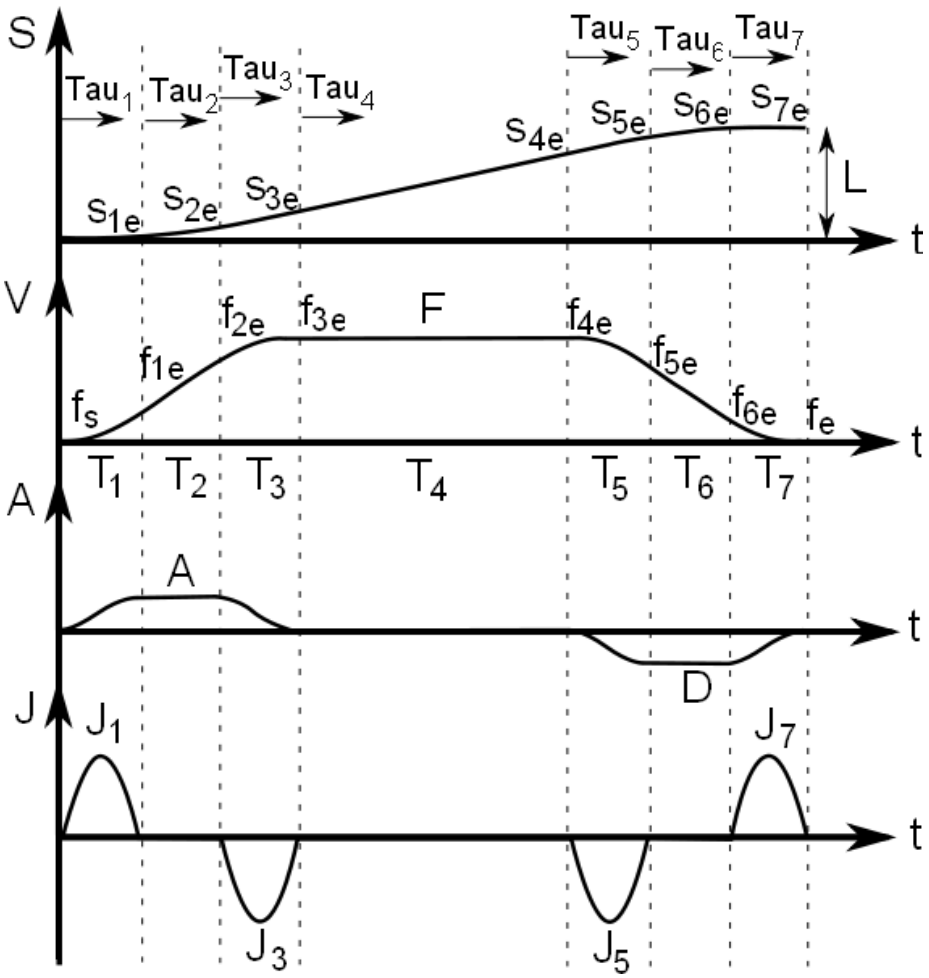
Acceleration:

$$\ddot{s}(\tau) = \begin{cases} -\frac{2A\tau_1^3}{T_1^3} + \frac{3A\tau_1^2}{T_1^2}, & 0 \le t < t_1 \\ A, & t_1 \le t < t_2 \\ \frac{2A\tau_3^3}{T_3^3} - \frac{3A\tau_3^2}{T_3^2} + A, & t_2 \le t < t_3 \\ 0, & t_3 \le t < t_4 \\ -\frac{2D\tau_5^3}{T_5^3} + \frac{3D\tau_5^2}{T_5^2}, & t_4 \le t < t_5 \\ D, & t_5 \le t < t_6 \\ \frac{2D\tau_7^3}{T_7^3} - \frac{3D\tau_7^2}{T_7^2} + D, & t_6 \le t < t_7 \end{cases} \tag{A.3}$$

Jerk:

$$\dddot{s}(\tau) = \begin{cases} -\frac{6A\tau_1^2}{T_1^3} + \frac{6A\tau_1}{T_1^2}, & 0 \le t < t_1 \\ 0, & t_1 \le t < t_2 \\ \frac{6A\tau_3^2}{T_3^3} - \frac{6A\tau_3}{T_3^2}, & t_2 \le t < t_3 \\ 0, & t_3 \le t < t_4 \\ -\frac{6D\tau_5^2}{T_5^3} + \frac{6D\tau_5}{T_5^2}, & t_4 \le t < t_5 \\ 0, & t_5 \le t < t_6 \\ \frac{6D\tau_7^2}{T_7^3} - \frac{6D\tau_7}{T_7^2}, & t_6 \le t < t_7 \end{cases} \tag{A.4}$$

A.2 Feed Profile Figure


S
Tau1
Tau2
Tau3
Tau4
Tau5
Tau6
Tau7
S1e
S2e
S3e
S4e
S5e
S6e
S7e
L
t
V
fs
f1e
f2e
f3e
F
f4e
f5e
f6e
fe
t
T1
T2
T3
T4
T5
T6
T7
A
A
t
D
J
J1
J3
J5
J7
t

References

[1] Y. Altintas and K. Erkorkmaz. Feedrate optimization for spline interpolation in high speed machine tools. Unpublished Journal Paper. 68

[2] Y. Altintas, K. Erkorkmaz, and W. H. Zhu. Sliding mode controller design for high speed feed drives. *CIRP Annals - Manufacturing Technology*, 49(1):265–270. 56, 57

[3] B. Bandyopadhyay and V. K. Thakar. Discrete time output feedback sliding mode control for nonlinear MIMO system: A stepper motor case. *International Journal of Systems Science*, 39(1):89, 2008. 11, 56

[4] K. Erkorkmaz. ME 780 - special topics in mechatronics - precision control systems, 2008. Graduate Course Notes. 34, 35, 56, 57, 60, 68

[5] K. (1) Erkorkmaz and Y. (1) Altintas. High speed CNC system design. II. modeling and identification of feed drives. *International Journal of Machine Tools and Manufacture*, 41(10):1487–1509, 08 2001. 33, 34, 35, 38, 42, 44

[6] Kaan Erkorkmaz and Yusuf Altintas. Quintic spline interpolation with minimal feed fluctuation. 2003 2003. 12

[7] Kaan Erkorkmaz, Chi-Ho Yeung, and Yusuf Altintas. Virtual CNC system. part II. high speed contouring application. *International Journal of Machine Tools and Manufacture*, 46(10):1124–1138, 8 2006. 13, 14

[8] A. Frank and A. Schmid. Grinding of non-circular contours on CNC cylindrical grinding machines. *Robotics and Computer Integrated Manufacturing*, 4(1-2):211–218, 1988 1988. 9

[9] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback control of dynamic systems*. Pearson, Upper Saddle River, NJ, 2010. 45, 47, 96

[10] K. Fujita and K. Sado. Instantaneous speed detection with parameter identification for AC servo systems. *Industry Applications, IEEE Transactions on*, 28; 28(4):864–872, 1992. 33

[11] C. L. Hwang, M. H. Wei, and W. J. Jieng. Non-circular cutting with a lathe using a three-stage intelligent controller. *Robotics and Computer-Integrated Manufacturing*, 13(3):181–191, 1997. 1, 9

[12] Joon Hyuk Kang, Chung Hyuk Yim, and Dong Il Kim. Robust position control of AC servo motors, 1995. ID: 1. 33

[13] B. Koninckx and H. Van Brussel. Real-time NURBS interpolator for distributed motion control. *CIRP Annals - Manufacturing Technology*, 51(1):315–318, 2002. 12

[14] Lennart Ljung. *System identification : theory for the user*. Prentice Hall PTR, Upper Saddle River, NJ, 1999. 34, 37

[15] Nick P. Manos. *Single controlled axis lathe mill*. University of Waterloo, Waterloo, Ont., 2002. 1, 2, 3, 9, 10, 15, 27, 67

[16] C. Nielsen. ECE 688: Nonlinear systems, 2009. Graduate Course Notes. 33

[17] F. Nollet, T. Floquet, and W. Perruquetti. Observer-based second order sliding mode control laws for stepper motors. *Control Engineering Practice*, 16(4):429–443, April, 2008 2008. 11

[18] C. Obermeier. Modeling of a permanent magnet disk stepper motor and sensorless field oriented speed control using an extended kalman filter. pages 714–20 vol.2, USA. (1)Lehrstuhl fur Feingeratebau und Mikrotech., Tech. Univ. Munchen, Garching, Germany. 11

[19] Mustafa Onat. Developing a PC- and SMS-mC-based stepper motor drive set. *International Journal of Engineering Education*, 24(1):23–21, 2008. 11

[20] Kandarp Patel. *Web based automatic tool path planning strategy for complex sculptured surfaces*. University of Waterloo, Waterloo, Ont., 2010. 67

[21] Gilbert Ka Bun Poon. *An architecture for CNC machine controllers*. University of Waterloo, Waterloo, Ont., 2005. 3, 10

[22] Tariq Samad. *Perspectives in control engineering : technologies, applications, and new directions*. IEEE Press, New York, 2001. 56, 57

[23] Yuwen Sun, Jun Wang, and Dongming Guo. Guide curve based interpolation scheme of parametric curves for precision CNC machining. *International Journal of Machine Tools and Manufacture*, 46(3-4):235–242, 3 2006. 13

[24] F. C (1) Wang and D. C. H. (1) Yang. Nearly arc-length parameterized quintic-spline interpolation for precision machining. *Computer Aided Design*, 25(5):281–288, 05 1993. 12

[25] M. Weck, A. Meylahn, and C. Hardebusch. Innovative algorithms for a spline-based CNC controller. *Production Engineering*, VI(1), 1999. 11

[26] W. Wilson. ECE 683 - system identification. 2008. Graduate Course Notes. 33, 34, 37

[27] Robert S. Woodbury. *History of the lathe to 1850; a study in the growth of a technical element of an industrial economy [by] Robert S. Woodbury.* Mass., Massachusetts Institute of Technology [1964, Cambridge, 1964. 1

[28] Stanislaw Zietarski. System integrated product design, CNC programming and postprocessing for three-axis lathes. *Journal of Materials Processing Technology*, 109(3):294–299, 2001. 9