MAKING CNC MACHINES SMARTER

by

Ilarion Kvitnevskiy

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement of the degree of

Master of Applied Science

in

Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2019

Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Abstract

CNC machines are a commonly used manufacturing tool. Over the years, they have become increasingly sophisticated. While there is a lot of research into making the machines more sophisticated, there is little research into making the machines smarter. CNC machines lack any intelligence to make decisions. Making a system fully intelligent is extremely difficult to do in one step. This thesis will focus on small steps that will hopefully lead to an intelligent CNC machine.

The thesis first explores using audio data for perceiving the cutting state of the machine. Experienced machinist can listen to the machine and determine how it is cutting and can assess changes for improving the cutting rate or surface finish. Ideally, the machine should be able to determine how it is cutting and use that information to adjust machine parameter for a cutting goal. In this project, a neural network was trained to detect the presence of chatter. Unlike conventional methods, this project involved only doing a Fourier transform of the audio data. The neural network had success in identifying chatter in the audio data in all the cases that were tested.

Next the thesis explores incorporating a model of the cutting process and using it to generate its own toolpaths. This method involves using a cutting model that uses 2D pixels for determining the cut and uncut area. Using this model, a tool path is generated by optimizing each step to achieve an optimal cutting goal. Further, constraints are added to the optimization, which improve the toolpath by limiting the turning radius, which makes the path smoother. The result is a toolpath that maintains a consistent cutting force, and smooth turning.

The previous project relied on a simplified model of the cutting process. As CNC machines become smarter, they will need to have more accurate models of the process. Part of this would be to have accurate dynamic models of the machine. The last project focuses on building an automated device for capturing such models. This device uses a novel approach compared to traditional tap testing. The devices uses a voice coil for actuation, a load cell for force measurement, and a laser displacement for measuring the vibrations. This

allows the tap tester to be able to measure many different tools without manually attaching accelerometers to each tool manually.

Acknowledgements

Thanks P.Bedi! I couldn't have done it without you.

Thank you everyone on #unmatlabic, in particular (without particular order):

@DarthShader

@aluminum

@watfor

I would also like to thank these members of #unmatlabic in particular (and in particular order):

@goose

@ubereng2

@Footnote

I could have done it without you, but your distractions were always amusing.

Thank you Connor Murray for being my rock climbing partner and getting me into lead climbing. I realized that my biggest fear is not this thesis, but heights.

Dedication

This is dedicated to Amanda for being my personal slave driver and keeping me on track to finish this thesis. Without you, I might have finished it 2 months late, instead of 1.

Contents

List of figures

List of tables

Graphic and Quote

“I love deadlines. I love the whooshing noise they make as they go by.”

— Douglas Adams, The Salmon of Doubt

Introduction

Background

Market

CNC machines are widely used for manufacturing parts. Their versatility allows them to be used for one off parts to mass production. In 2018, the global CNC market was valued at USD $63 billion [1], and is expected to grow 7% each year. In Canada alone, machine shops produce $6 billion in revenue yearly [2]. The most common CNC machines are lathes and mills. They are used for cutting raw material into the desired shape.

CNC milling machines and operations

The most versatile CNC machine is the CNC milling machine. The raw part is put on a table that is moved by motors. The raw part is then cut by a tool called an endmill. Unlike a drill bit, an endmill can cut sideways. Using this method, different shapes can be cut into the raw material to create the desired part. CNC machines were developed and popularized in the 1960s - 1980s. G-Code is the code that is used to control such machines. It is a series of positional instructions - 'Move to X coordinate at F speed'. A CNC machine executes and follows the given toolpath to produce the part.

A typical operation of a CNC milling machines follows as such:

1. Given a 3d model, a CNC programmer uses CAM software to generate a toolpath.
2. The CNC machine is loaded with the raw material, toolpath, and necessary tools to make the part.
3. The machine is then run by an operator. An operator usually oversees multiple machines running at once.

4. Once the part is finished, it is unloaded from the machine, and the next part is loaded.

Programming the machine requires knowledge about the machine, and how it behaves under different cutting conditions. The machine can cut the part in different ways with different tools, and it is up to the programmer to make these decisions.

While the machine is cutting, an operator must be vigilant of the cutting conditions inside the machine. A common problem that the machine can experience is chatter. When a mechanical system is excited it will vibrate at a certain frequency. The endmill, tool holder, spindle and the rest of the CNC machine is a complicated mechanical system. When it is excited, the system will vibrate. When the machine is cutting, there is a force that is applied to the endmill, and since this force is periodic it will excite the system. In certain scenarios, the cutting force will be at a specific frequency which will lead to a buildup in vibrations, which is known as chatter [3]. This increases the forces on the endmill, which wear out the tool, and can cause it to break. The operator can override the machine parameters (feeds and speeds), to reduce the chatter. In the case of a broken tool, the operator needs to stop the machine and replace the tool. In addition to chatter, there are many other situations the operator must look out for, such as chip build up, and worn tools.

As sophisticated as they are, CNC machines are not intelligent. A CNC machine does not make any decisions, and it has no sensors to monitor how it is operating. All of these decisions are made by the programmer and operator. They have to use past experience to decide what to do. This knowledge is specific to the application - the machine, material, tools, etc. Furthermore, conditions while cutting can change. The tools get worn out, the material is not uniform. Therefore, programmers choose safe and conservative parameters to account for this variability. In industry there is no real-time monitoring - it requires a human to intervene if something goes wrong. When mistakes happen, they can be expensive. The machine can break the part that was being manufactured, causing it to be scrapped and remade.

The machine can also be damaged, in which case it cannot be used to make parts until it is fixed. This can slow down production and cause financial strain on the manufacturer.

The job of looking after the CNC machine is labor intensive, require experience, but is dull.

Definition of terms and jargon

TABLE 1 TERMS AND DEFINITIONS

Term	**Definition**
Endmill	Cutting tool used in mills. Looks like a drill bit, but can cut sideways. Many different types and shapes exists for various applications.
Tooth/flute	Endmills have flutes (like drill bits). Each flute makes contact with the material it is cutting. More flutes means that in one rotation of the endmill, each flute will cut less material. In the case of endmills with inserts, the cutting inserts are referred to as teeth.
Table	The part of the mill that moves. The vise and stock is attached to the table.
Stock	Raw material that will be used to create the part.
Spindle	The rotating component of the machine. In a mill it holds the endmill.
Toolpath	A trajectory that a CNC machine follows to create the part.
Programmer	A person that creates toolpaths either using CAM software or by hand.
CAM software	Computer Aided Manufacturing - software for helping generate toolpaths.
Feeds	The terms used for describing how fast a machine is programmed to move.
Speeds	The spindle rotation velocity.

Depth of Cut	Depth of the cutting tool in the raw material, measured along the tool rotation axis.
Width of cut	Width of the cutting tool in the raw material, measured perpendicular (radially) from the axis of the tool rotation.
Vise	A device used for clamping and holding the stock.
Vibrations	Periodic displacement of the machine. These displacement are small, and are typically the highest at the tool tip.
Cutting forces	Forces generated on the endmill when it is cutting material. Due to the rotation of the endmill, the cutting forces are periodic, and cause vibrations.
Chatter	When the machine resonates. Typically because the cutting forces are exciting the resonant frequency of the machine.
Cutting frequency / tooth passing frequency	Spindle speed multiplied by the flute count. This is how often the material is cut by a flute.

Why make a smarter CNC machine

Machines for manufacturing parts have been around for centuries. Over time, they have become more and more sophisticated. Milling machines are one of the most common and versatile machines. Manual milling machines require a machinist to make each part. The machinist must plan how to make the part, what tools to use, how to hold it. To make multiple parts, he has to repeat the same process multiple times. Using a computer to control the machine allowed for higher precision, complex shapes, and more efficient use of labor. The machinist can now figure out how to make the part once, and the machine will make as many parts as needed. However, the machine still needs to be monitored. This job is dull. The job is complicated and broad enough that it cannot be automated away easily, but at the same time, it is repetitive and never changes.

The inefficiencies of current CNC machines is the lack of quantitative data that the machines provide. There is a lot of qualitative data - this part turned out good, using these parameters made the part faster, this cut sounded good - but qualitative data has to be interpreted by a human, and has limited usefulness. By how much should the feeds increase, should the depth of cut be increased or width of cut, when should this tool be replaced? All of these questions cannot be answered using qualitative data. To optimize the time to make a part, a programmer has to program the part, make it, observe how it was made, make judgements about how to tweak it, and then make it again. Such an iterative process is a waste of both human and machine resources. Instead of making parts that make money, the machine is making scrap parts. And instead of programming new parts, the programmer is tweaking the same part over and over again.

By making CNC machines smarter, they can provide qualitative data which can be used to increase the efficiency of the machine and also free up machinist to do other tasks.

First steps

Ideally a smart CNC machine would behave as such. The raw material would be loaded automatically by a robotic part. The machine detects where the part is located automatically using sensor. The 3D geometry of the part to be made is sent to the machine. The machine determines the best tools to use and generates a toolpath. A simulation of the entire cutting process is generated. The machine starts making the part and is monitoring the cutting in real-time with various sensors. If the sensors measurements do not line with the simulation of the cutting, the CNC machine takes corrective action and updates its system model. Once the part is done, it is checked that it is within tolerance and is automatically removed.

The above description of smart CNC machines is too ambitions for a single thesis. Instead, this thesis focuses on a few smaller aspects. An autonomous system must know its cutting state to answer questions such as “can the material be removed faster”, “can the cutting speed be increased”, and so on. Next, an autonomous system must have the ability to alter the tool path to a purpose. So in this thesis, for the second step, a path planning strategy is devised that uses a model of the system. The path is made according to the current state of the system and can change if the system changes. The third step would be getting an accurate dynamical model of the system. An automated system needs a model that can predict the result of taking an action. A device is created that will automatically measure the dynamic model of a CNC machine.

Monitoring the machine using audio data and ML

Experienced human CNC operators know if a machine is cutting properly using just their hearing. The machine cutting process creates a lot of vibrations, which are audible. This noise contains a lot of information. A new sharp endmill makes different sounds compared to an old dull endmill. Chatter has its own distinct sound. An operator learns over time how to decipher the noise of a machine into meaningful metrics. They can monitor a machine, and preemptively take action before a catastrophic failure happens. They can hear a dull endmill, and swap it out for a new one, before it breaks and causes damage.

Neural network research has evolved to a level where it can recognize information embedded in a voice signal. This thesis proposes to explore the use of neural networks for detecting machining state for use on a small CNC machine. The goal would be to use just a microphone and a computer to detect if chatter is happening in a machine. The benefits of this approach is that it is noninvasive (nothing about the machine has been changed), and it can be improved as more data is collected.

Constraint based toolpath planning

Toolpath planning currently involves human planning. Picking feeds and speeds, picking tools, depth of cut, and other parameters. Usually this is done by an experienced machinist. The machinist uses past experience to pick parameters. Furthermore, software that generates the toolpaths requires a lot of manual input. The programmer selects the area, the toolpath strategy, a tool, cutting parameters, and many other details related to the toolpath (e.g. how the endmill enters and exits the material). This allows the CAM software to generate a toolpath for the selected area. Although a far cry from manually writing GCODE it is still labor-intensive process. Furthermore, it is not clear the relationship between the parameters and the actual limits of the machine. For example, for a roughing operation, the goal is to remove as much material as fast as possible. The limit is the cutting force on the tool and its deflection - if there is too much force on the tool, it will break. The parameters that the programmer selects are speeds and cutting depths. They have a non-linear, and non-intuitive relationship with cutting force. A programmer has to pick parameters that will generate a toolpath that will not exceed the cutting force limit. This requires the programmer to have a lot of experience with the machine, tools, and material that is being machined. Furthermore, optimizing the tool path requires making a part, observing where there are inefficiencies, and re-programming the part. This is repetitive and dull labor prone to errors due to boredom.

For this project, an automatic toolpath generator is investigated. The generation will use a cutting simulation. In this simulation a tool path can be will be generated step by step. Each step can be evaluated and compared against other steps. Using this method, the most optimal step can be taken to achieve an optimal toolpath.

For this approach to be useful, the simulation must be accurate. For this project, a simplified simulation is used to verify that the method works. For more practical application, a sophisticated model that captures the dynamics of the system is needed.

Automatic tap tester

The dynamics are used to model the machine as a structure that can deform. The dynamics are represented as frequency response functions. This represents the machine's stiffness at different frequencies. Since the cutting process is periodic, it can excite many different frequencies. Depending on which frequencies are excited, the machine and the tool being used can chatter. The cutting process can generate frequencies that excite the natural resonances in the machine. This creates vibrations in the tool and the tool deflection leads to increase in the maximal cutting force. High tool deflection leads to tool wear, out of tolerance parts, and broken tools. Chatter can be avoided by picking the correct parameters. These depend on the machine dynamics. Knowing the machine dynamics, a model of the machine can be made. This model can predict chatter, and tool deflection. Combined with known machine parameter - such as power, torque, and the tool geometry, a comprehensive model of the cutting process can be made. This model can be used for planning much more efficient toolpaths, and automatically picking the best tools to use.

However, the dynamics depend on not only the machine, but the tool and the tool holder. Picking the correct parameters can increase material removal rate by several times.

To determine machine dynamics the most common method is using tap testing. This involves mounting an accelerometer on the tip of the endmill and hitting the endmill with a hammer that can measure force. The impact excites the system, making it resonate. The accelerometer measures these vibrations, and the data can be used to a frequency response function.

However, tap testing is time consuming and is not commonly used in industry. The equipment is not cheap and requires know-how to use properly. There are many different ways to perform the tap test, and each have their own downsides.

An automated method for performing a tap test is proposed and tested in this thesis. Instead of mounting an accelerometer and using a person to tap the machine, a laser displacement sensor, and a voice coil to tap the endmill are proposed. This would requires that the device be set in the machine once, and all of the tools can be tested automatically.

Thesis Layout

This thesis has 3 main projects. Each project will have its own chapter. Each section will begin with an overview of the current state of the industry and research that has been done in that area. There will be a section discussing the background knowledge related to the project. Following the background, the project scope will define what problem will be solved, and how it will be approached.

With the background established, the main section will detail the development, and proof of concept for each project. This will be followed by results, an analysis of the potential applications, and limitations. Lastly, the section will close off with a discussion for future improvements of the project, and what were the failures of the project.

Monitoring the machine using audio data and ML

Current state of affairs

Industry

There are a few commercial monitoring systems that allow the detection of chatter using a microphone. One such system is ChatterPro by MALinc. The software needs to know the spindle speed and tooth count on the endmill. Using this, the software looks for frequencies that are not harmonics of the cutting frequency. If a non-harmonic frequency is above a certain threshold, it is deemed chatter. The software can interface with the CNC controller to then increase the spindle speed until chatter goes away.

While this solves the issue of monitoring chatter, the software requires the knowledge of tooth count of the endmill, and the current spindle speed. Unfortunately, there are no examples of this software being used in production that can be found online. In production, a CNC machine will use multiple endmills, and there is no way to get that information from the CNC machine to the software.

Aside from chatter monitoring, there is no other type of monitoring done using audio. For example, a common problem that does not allow machines to run unsupervised is tool breakage. CNC machines have no way to know if the tool they are using is broken or not. Sometimes the entire tool is not broken, but just one flute. All these problems are easily heard and identified by an operator.

Research

Vibrations in machining have been studied for over a century. The classical approach is to model the machine as a linear dynamic system [3]. The machine is simplified to a structure of multiple mass-spring-dampers. While the machine is cutting material, it deflects and leaves a wavy surface finish behind. The next tooth starts cutting this wavy material and is further excited. If the wavy-ness of the surface matches with a resonant frequency of the machine, each new cut will excite the endmill more and more. This will cause the deflections to increase over time. For a simplified cutting model, it is easy to calculate cutting conditions that will cause the machine to chatter. A stability lobe diagram (SLD) shows which spindle speeds and depths of cut cause chatter. However, it becomes more complicated to calculate these conditions for more accurate models that fully include the shape and geometry of endmills. Nonetheless, even simplified models produce accurate results and are useful. The issue is that to create a stability lobe diagram, the machine dynamics need to be known. Modeling a machine is not easy and requires specialized knowledge and tools.

There has also been research into predicting chatter real-time. This requires a sensor to monitor the machine. Accelerometer and microphones are common sensors since they can be placed inside the machine without changing the cutting process. Riviere [4] summarized three common techniques used for detecting chatter using audio.

1. Use a threshold for the amplitude of audio. Chatter is louder than normal cutting conditions. This is a simple method but require a very carefully selected threshold.
2. Analyze the audio in the frequency domain and set a threshold. This requires removing spindle and tooth passing frequencies and their harmonics.
3. Sample the audio at the spindle frequency. If there is no chatter, the audio level should be identical every time the spindle angle is in the same position. If there is chatter, the audio amplitude will vary. This is very simple but assumes that the cutting process is uniform.

In addition to these methods, Cao [5] implemented a more sophisticated analysis. His approach was as follows:

1. Wavelet transform of the audio. Wavelet transform is similar to a sliding window Fourier transform, except that it has better accuracy for transient (non-periodic) data.
2. Remove spindle and tooth passing frequencies and their harmonics.
3. Perform SVD (singular value decomposition), to transform the data from frequency vs time to components vs time. The first axis (component) of the transformed data is assumed to be chatter. This data is now one dimensional (chatter vs time).
4. Set a threshold on the chatter component.

This approach uses a statistical procedure to transform the data from the raw collected data into a direct metric that is useful.

In recent years machine learning has become a widely used tool. It is a statistical model that for recognizing patterns in data. There has been some research in using machine learning for chatter recognition. Most of the research focuses on estimating the stability lobe diagram using machine learning, it is not useful for real-time monitoring of the machine.

Cao [6] has also proposed a method for training a neural network to identify chatter. His strategy involves the following:

1. Remove the spindle and tooth passing frequencies and their harmonics.
2. Extract features from the time and frequency domain. These are statistical metrics such as average, energy of signal, and non-linear features.
3. Train a self-organizing map neural network. The network takes in the features and an input, and outputs its locations on a map. Location nearby are considered similar, and location far away are considered not similar.

4. Deviation from non-chatter cut on the map is the 'severity of chatter' metric. Once this metric passes a certain threshold, chatter is detected.

This approach is effective at detecting chatter and can detect it before visible chatter marks appear on the part.

All the methods mentioned have several similarities. Initially, all the methods process the audio and remove the spindle and tooth passing frequencies and their harmonics. While this is known in a test setup, in production the CNC controller might not be connected to a monitoring system. All of the approaches also used one endmill to test chatter detection. Furthermore, all of the approaches required manual thresholding. This means that every combination of tool and machine might need a different threshold. This means that the methods still require manual work to properly work in all conditions. This would limit the usability of these approaches in industry.

In addition to that, these approaches focus heavily on detecting chatter, and do not consider any other type of conditions that might be affecting the cutting process such as a dull endmill.

This project will extend on the ideas and will investigate using machine learning for detecting chatter.

Project definition

The weaknesses of the discussed literature will be the focus of this project. The project should be able to accomplish chatter detection without the use of manually setting thresholds and manually processing the data. The method should be work with multiple tools, even with different flute counts, which should mimic a human CNC operator – a human CNC operator does not need to be told what the spindle speed or the flute count is.

To summarize:

- The data cannot be modified (removing spindle frequencies), but can be transformed (Fourier transform) for usage in a neural network.
- The detection algorithm should work with multiple endmill and different flute counts.
- There should be no manual threshold setting. The system must work without being tuned by a person.

Theory

In statistics and machine learning, data is represented as a multidimensional vector. Each dimension represents some sort of variable. For example, in an image, each pixel would correspond to a dimension, and the amount of dimensions corresponds to the amount of pixels. Every unique image can be represented as a unique vector. The goal in machine learning, is to identify areas in the multidimensional space that represent certain categories.

The simplest approach would be to use simple linear classifiers. These separate the data using a hyper-line. However, most data cannot be separated using a line. The data is non-linear, and is spread apart in the multidimensional representation. However, it is possible to transform this data into something more usable using neural networks.

Neural networks are brain-inspired systems that can be trained to perform tasks. They consist of perceptrons connected together is a network. Each perceptron takes in many inputs and has one output, similar to a neuron. Each input is multiplied by a weight and is summed together with a bias. This output is then the input to an activation function. The output of this function is the output of the perceptron. Figure 1 shows the configuration of a neuron.

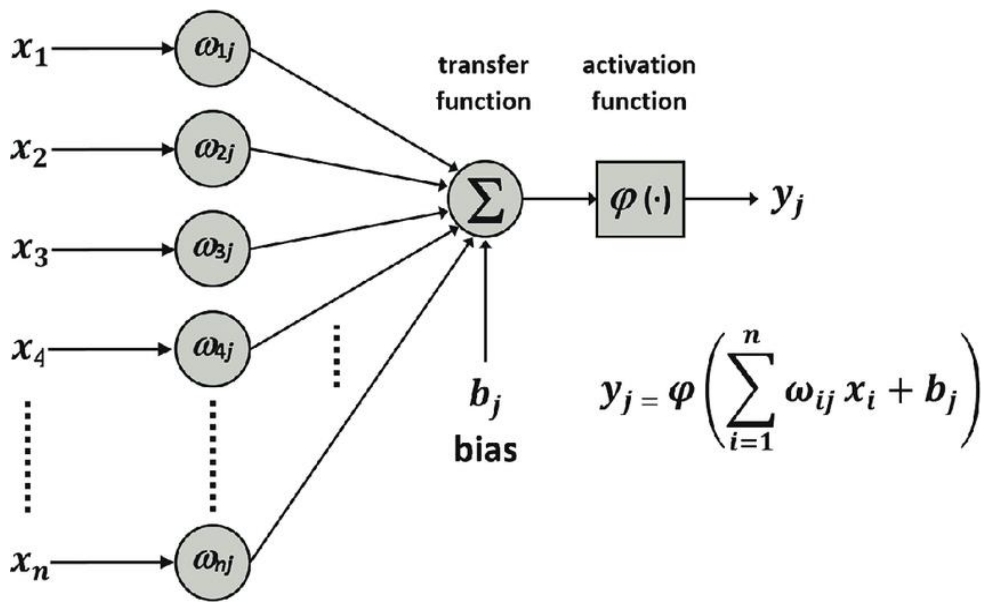


FIGURE 1: PERCEPTRON STRUCTURE [7]

Multiple perceptrons are connected to form a network. Depending on the configuration, the network can simulate any possible function. Neural networks can be thought of as a more advanced version of ‘curve fitting a function’. Classical optimization problems have few inputs, and have closed form solutions. Neural networks are useful when the problems have many inputs and the functions they are simulating are complicated. Below is a diagram showing an interconnected neural network. Neural networks are trained by using known inputs and outputs. The weights of the neurons are changed progressively to make the neural network predict more and more accurately.

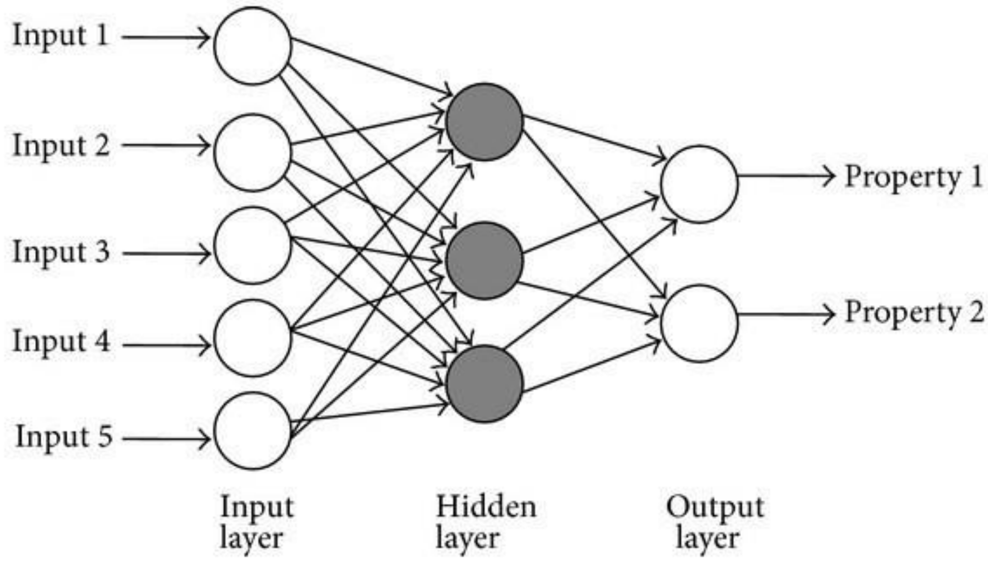


FIGURE 2 MULTILAYER NEURAL NETWORK [8]

If the activation function is an identity (i.e., the output is the input), the perceptron becomes linear. The input data is multiplied by the weights and is added together - a dot product. The outputs of the first layer is combined and becomes the input for the second layer - another linear operation. These operations can be combined together and simplified as a matrix transformation. This limits the function that the neural network can simulate to linear transformation. If a non-linear activation function is used, then the neural network can simulate non-linear functions. Using a nonlinear activation functions can be thought of as non-linear transformation.

An autoencoder is a special configuration of a neural network that is used to reduce the dimensionality of the data. The input and output layers are the same, and there is a hidden layer(s) with fewer nodes than the input. The network is trained to reproduce the input as the output. The training of this neural network results in an encoder, and a decoder. The encoder is the first half of the network which encodes the data down to the specified number of dimensions. The second half is the decoder, which takes the reduced data, and reconstructs the original data.

The utility of the autoencoder is to reduce the dimensionality of the data. Often, the data has so many dimensions that it is difficult to create patterns from it. However, when the data is compressed to fewer dimensions, patterns emerge. Furthermore, the autoencoder can also be used to remove noise from data. Since noise does not have a contribution to the information in the data, it is not encoded. Thus, encoding, and then decoding the data removes noise.

Implementation

This project will build upon Cao's idea to use singular value decomposition to transform the data. SVD is a linear transformation that can be replicated by a linear neural network. Expanding on this idea, this project will attempt to use a non-linear neural network on the unmodified data. The data input to the neural network will be frequency data, and the output will be dimensionally reduced data. This output data can then be used to determine chatter using a simple classifier.

An autoencoder was chosen for the architecture of the network because of its versatility and ease of use. A neural network with only one output is harder to train properly, since there are more layers to reduce down to only one output. Furthermore, the reduced dimensionality will show patterns in the data, and allow for easier visualization.

The project is divided into 4 parts:

1. Data acquisition
2. Audio preprocessing
3. Feature extraction
4. Classification

Data acquisition

The experimental setup uses a desktop CNC milling machine and a microphone mounted on a boom close to the work piece, see Figure 3. The CNC machine is a converted manual Taig machine. It has a fixed speed motor, and pulley for setting the spindle speed. The machine performed test cuts while the audio was recorded to a laptop computer.

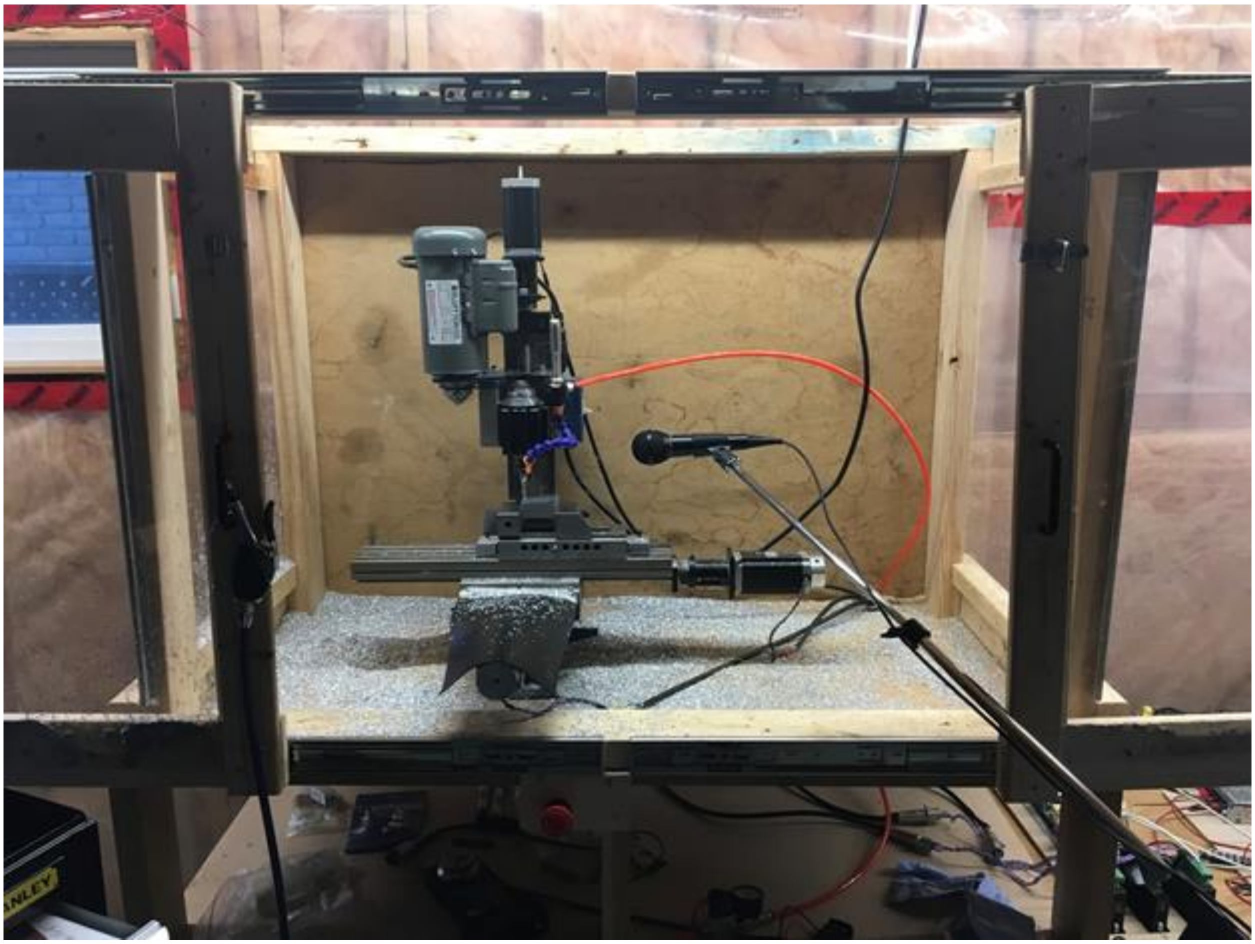


FIGURE 3 EXPERIMENTAL SETUP

Test cuts

The test cuts were a series of cuts on a block of aluminum. The cuts varied four different parameters: width of cut, type of endmill, spindle speed, and feed speed. The first three parameters will change if chatter happens or not, however, the feed speed only changes the amplitude of chatter. Each test consisted of varying the parameters that cause chatter. Each test was run at various feed speeds which increased the amount of data that can be used for classification. A total of 30 minutes of audio data was recorded.

In the recorded data, there were a total of nine tests. Each test was then manually categorized as 'chatter' or 'good cut' by listening to the recorded audio. Each test case used a 1/4 inch endmill, either 3 flute, or 2 flute. The machine had three possible spindle speeds - 2600 rpm, 4200 rpm, 6700 rpm. Each test case was run at 160 mm/second and 192 mm/sec. The width of cut was 100% (0.25 inches) and 70% (0.175 inches). The depth of the cut was 1.1mm for all test cases.

TABLE 2 TEST CASES AND RESULTS

Test No.	**Flutes**	**Spindle**	**Width of cut**	**Chatter**
1	3	4200	70%	no
2	2	4200	70%	yes
3	3	2600	70%	yes
4	2	2600	70%	yes
5	2	2600	70%	no
6	2	4200	100%	yes
7	3	2600	100%	yes
8	3	4200	100%	no
9	3	6700	100%	no

Audio preprocessing

The sound is created from the flutes of the endmill cutting into the work piece. The spindle speed in the tests performed can go up to 4200 rpm, and the endmill can have up to three flutes. This means that the cutting frequency will be at 210 Hz. There will be harmonics of this frequency, and other chatter frequencies in this range. Thus, a bandwidth of 2000 Hz is sufficient to capture the information in the sound. The audio was recorded at 44.1 kHz, which is the standard sampling rate for a computer microphone. It was then down sampled by a factor of 10 to 4.41 kHz.

The next step is to perform a Short Time Fourier Transform (STFT). A Fourier transform takes a signal and outputs which frequencies are active. A STFT takes progressive chunks of the signal and applies a Fourier transform. The transformed signal shows which frequencies are active during what time. A STFT plot is called a spectrogram. The X-axis is the time axis, the Y-axis is the frequency, and the color represents the amplitude of the of the frequency. Figure 4 below shows the spectrogram of the collected audio. It only shows the first 500 Hz, since that is where the majority of the information is.

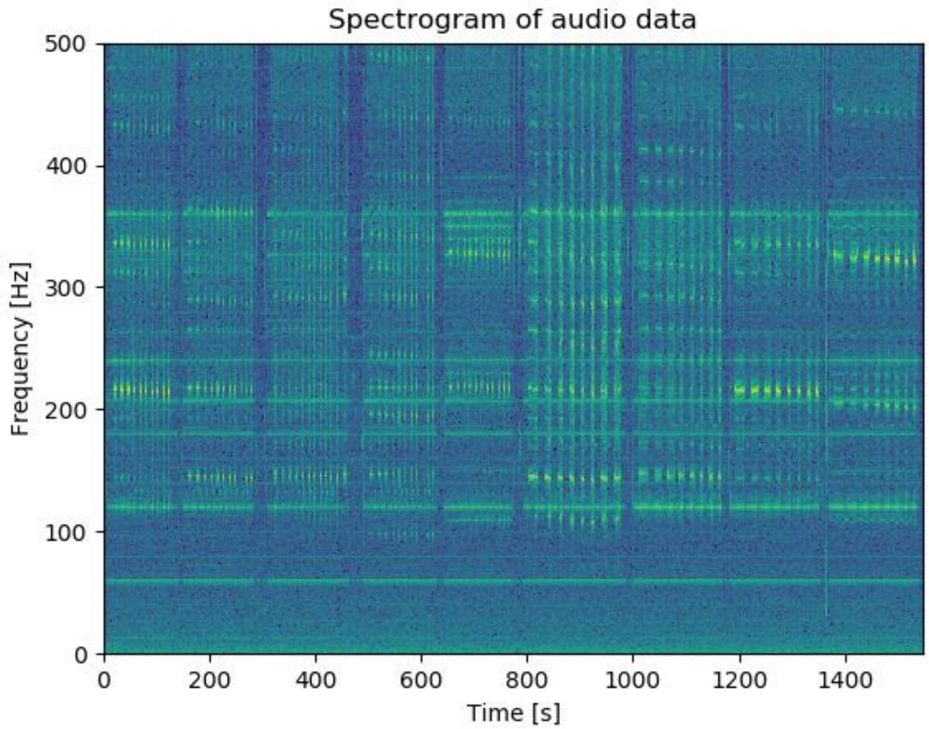


FIGURE 4 SPECTROGRAM OF AUDIO DATA

In the spectrogram above there are nine different tests. Each test has cuts at various speeds. There are certain bands that are active during the cutting. Some of these correspond to cutting, or chatter. Others are background noise: compressed air, or the stepper motor noises. Figure 5 below shows the spectrogram for a single test. Each test has several cuts in it. Each cut is visible in the figure. It is also possible to see the motor spinning frequency and the cutting frequency.

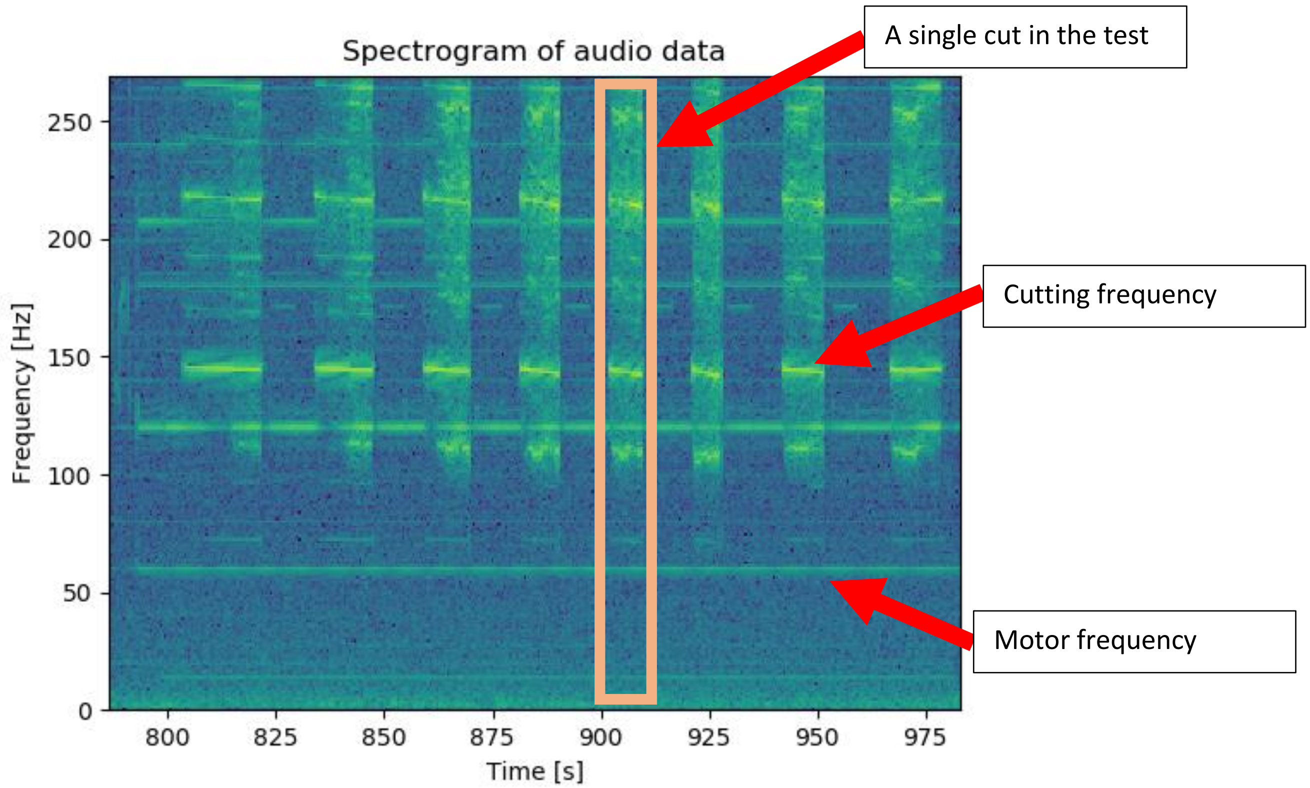


FIGURE 5 ZOOMED IN SPECTROGRAM OF ONE TEST

Feature extraction

Each time instance is considered a separate data point. Thus, each data point is a vector that represents the spectral density. That is, each dimension of the vector represents a particular frequency range. The spectrogram data contain a little bit more than 2000 frequency ranges.

Autoencoder

There were three main autoencoder parameters that were tuned for this project: activation function, dimension to reduce to, and number of hidden layers. These parameters were tested with the data, using 70%/30% split for training and validation. The loss was comparing the original data to the data after being encoded and decoded. Since the autoencoder does not know which data points are chatter or not chatter, the autoencoder will not try to encode to a dimension that distinguishes chatter.

Activation functions

Each neuron in the autoencoder uses an activation function. Four commons activation functions are: linear, sigmoid, 'RELU', and 'SELU'. RELU is like linear, but all negative values are mapped to zero. SELU is similar to linear, except that all negative values are map to an exponent. The advantage of SELU is that its derivative is continuous, which is helpful for back propagation [3]. The sigmoid function was not appropriate for the data, since the range of function is from 0 to 1, and the data can be an unbounded number. The data was not normalized, because potential future data could be even louder, thus, it would get clipped.

The remaining three activation functions were tested against each other; the result is below in Figure 6. RELU performed significantly worse than SELU and linear, with SELU performing slightly better. Both SELU and linear have continuous derivatives. This enhances the performance of back propagation and provides better results. For the rest of the project, SELU will be used.

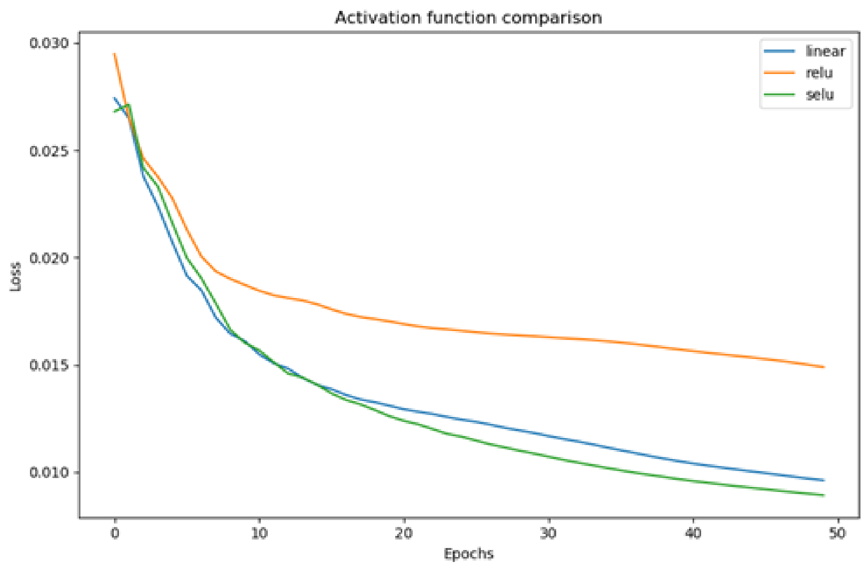


FIGURE 6 ACTIVATION FUNCTION COMPARISON

Reduction dimension

The number of dimensions impacts the fidelity of the reconstructed signal. The data was compressed to 2, 4, 6, 8 ,16, and 32 dimensions, as show in Figure 7. Unsurprisingly, compressing the data to fewer dimensions increases the loss. What is surprising is how well the data compresses. Using only two dimension has a loss of 1.5% but using 16 dimensions has a loss of 1%. This is a not a significant change in loss. Furthermore, when the number of dimensions doubled, the error increased approximately linearly. This shows that there is not a diminishing return from using fewer dimensions.

Using two dimensions has the advantage that the encoded data can be plotted on a XY scatter plot. This allows easy visualization of the data. For this project, only 2 dimensions were used for the encoder.

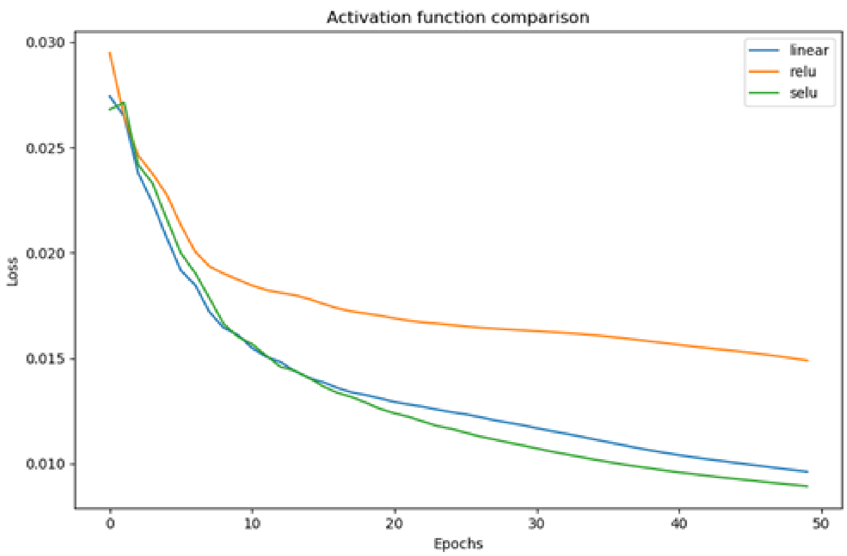


FIGURE 7 HIDDEN LAYER DIMENSION COMPARISON

Number of hidden layers

The autoencoder can have multiple hidden layers, each progressively having fewer nodes. The autoencoder was tested with 1 to 4 hidden layers. For simplicity sake, when having multiple layers, each layer would reduce by the same factor (e.g. 2000->200->20->2). The results can be seen in Figure 8.

Using 1 or 4 hidden layers produces significantly worse results than using 2 or 3 layers. When using one layer, there are only two neurons connected to 2000 inputs. This is in efficient for back propagation, since each one of the inputs directly changes the output. There are many local minimums, and the training is not effective. In the case of 4 layers, there are too many layers between the input and output. There are 4 layers to encode the data, and 4 more layers to decode the data. Furthermore, each of the connections has a weight. This means the number of variable in the neural net is high. With this many layers, the data can be 'forgotten' in the neural network. Since there are so many transformations in the network, the data becomes different from the original. Thus, back propagation is once again not effective.

For the data used, the sweet spot is 2 or 3 hidden layers. They both have similar loss well, but 2 hidden layers were faster to train.

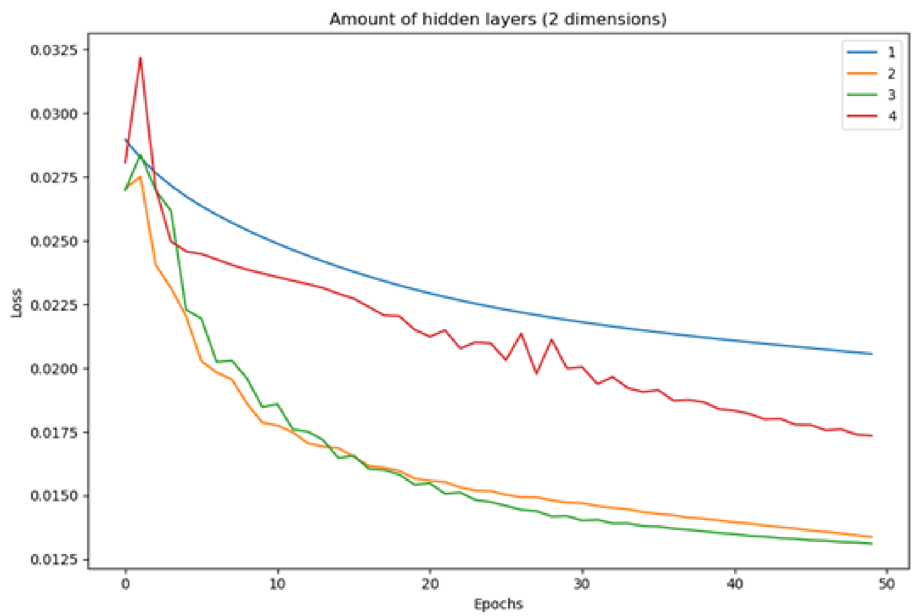


FIGURE 8 NUMBER OF HIDDEN LAYERS COMPARISON

Autoencoder results

From the previous experiments, the optimal parameters were selected. The autoencoder had 2 hidden layers, 2 dimensions to reduce to, and used SELU as the activation function. The results of the encoded-decoded data are compared to the original data in Figure 9 below.

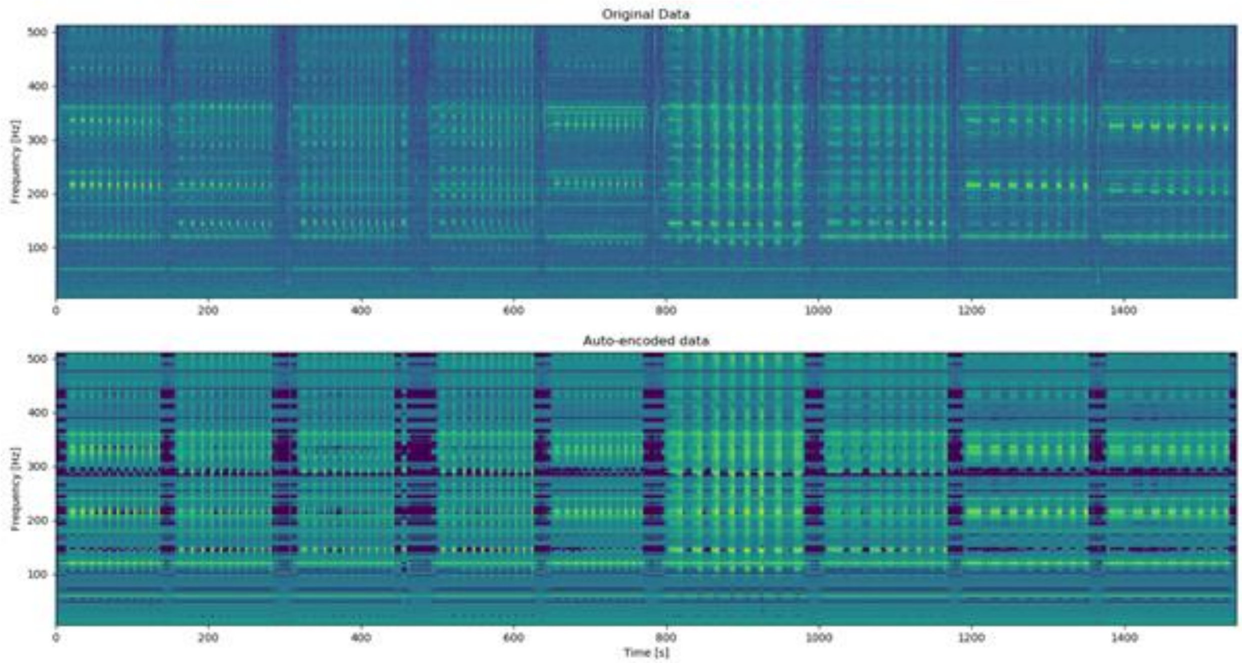


FIGURE 9 AUTO ENCODED DATA COMPARED TO THE ORIGINAL DATA

The auto encoded data contains all the major frequency bands from the original data. Furthermore, the autoencoder removes noise from the data, which can be seen between the test cuts. In the original data there is 'static' looking white noise, but in the auto encoded data there is no noise.

Classification

Now that the data has been encoded to 2 dimensions, it is possible to visualize it, and pick a classification strategy. Each time the autoencoder was trained, the results would be different. This is because the weights are randomized, and there are many local minimums. A neural network does not guarantee the global minimum solution. Two examples of the auto encoded data are shown below in Figure 10, and Figure 11. The red dots represent 'good cuts', the blue dots are 'chatter', and the black dots are 'not cutting'.

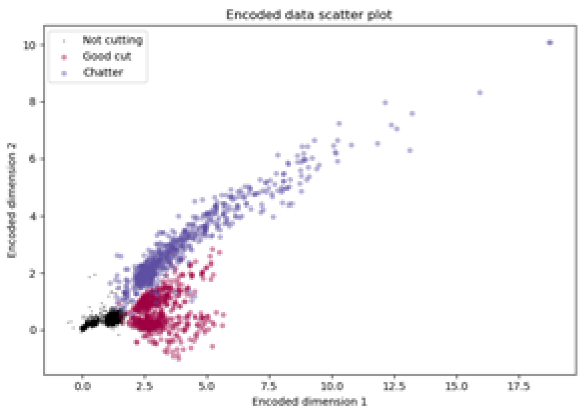


FIGURE 10 EXAMPLE OF ENCODED DATA

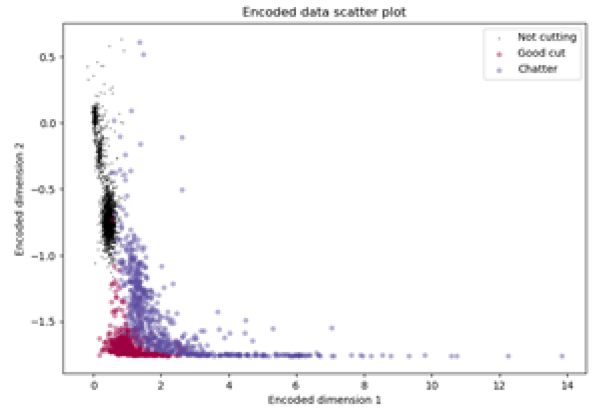


FIGURE 11 EXAMPLE OF ENCODED DATA

For classifying the data, three common classifiers were considered: Maximum likelihood, nearest neighbor, and support vector machine. Maximum likelihood classifier performs poorly on this sort of data set because the clusters are not Gaussian, they have shapes and curves to them. A nearest neighbors classifier can create complicated boundary geometry, that is unnecessary. The boundary geometry is simple, and a nearest neighbor algorithm is computationally intensive. A support vector machine would be the best candidate; it defines a line that is the boundary between the two classes.

The autoencoder sometimes encodes the data such that the chatter zones and not chatter are overlapping, and sometimes they are not overlapping. There is no reward for the autoencoder to have them separate. Thus, to achieve an encoding that separates the clusters, multiple instances of an autoencoder must be trained, and each instance evaluated. For the evaluation, a linear SVM classifier was used, and the autoencoder with the minimum error was selected. Minimum error is the autoencoder with an SVM that has the least misclassifications. Figure 12 below shows the optimal encoding for the data. The 'good cuts' and 'chatter' are distinctly separate. The direction of the cluster seems to be related to the type of cut in this encoding. Data points near the horizontal axis are 'good cuts', while being near the diagonal is a 'chatter cut'. The distance from the origin is related to the intensity of the cut, the origin being not cutting at all. When the machine is chattering, it is typically loud, and this is visible in the data: the chatter cuts have a large spread.

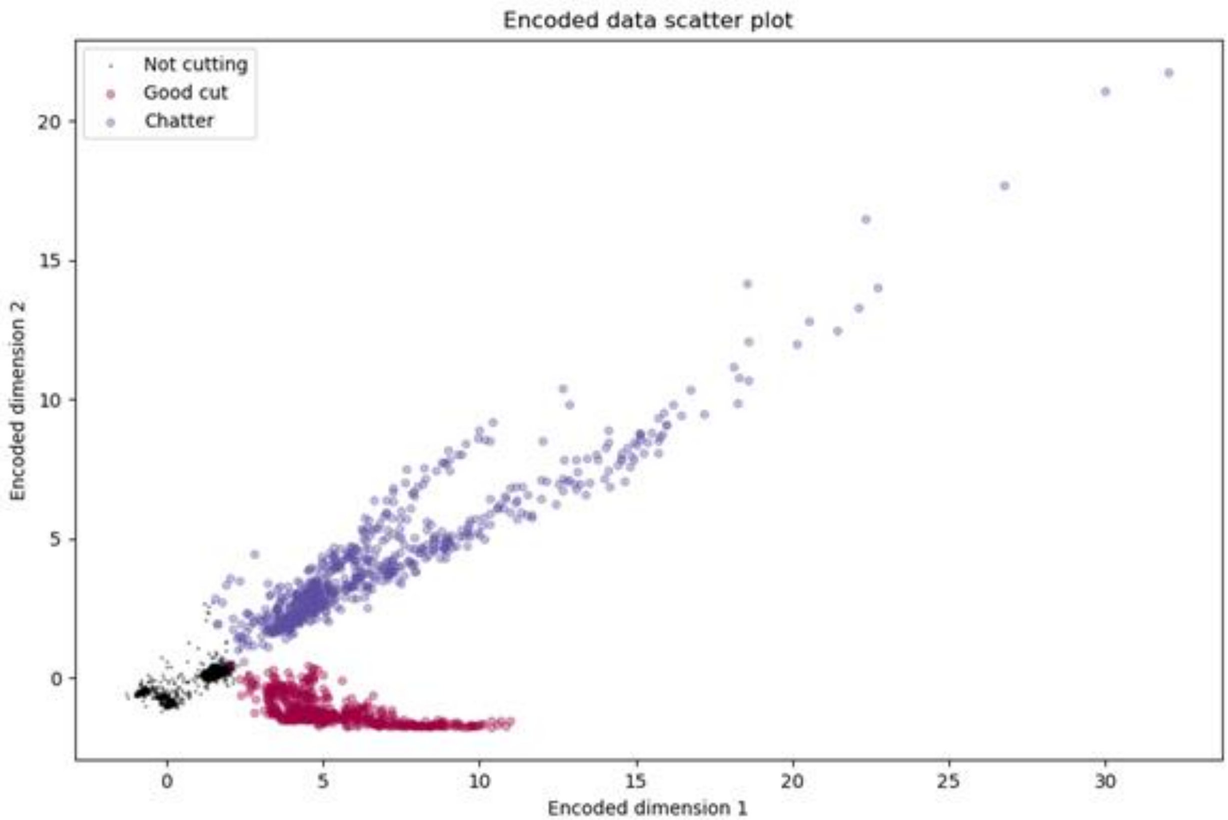


FIGURE 12 OPTIMAL AUTO ENCODED DATA

The linear SVM classifier for the optimal data set is shown in Figure 13. The linear SVM only miss classifies one data point. This shows that it is feasible to classify chatter using an autoencoder and a linear SVM classifier.

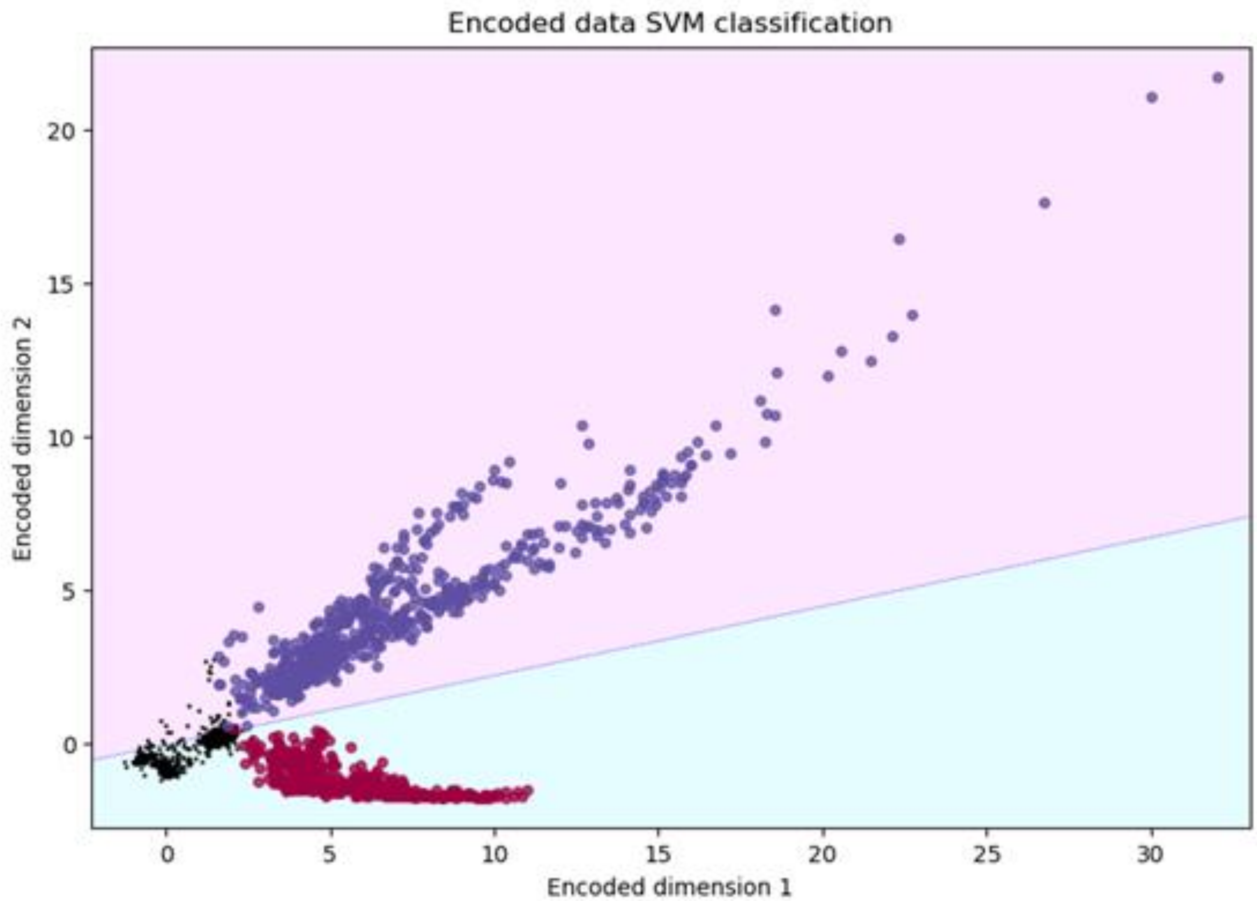


FIGURE 13 SVM CLASSIFICATION OF OPTIMAL ENCODED DATA

Results

The project has shown that it is possible to create a system that identifies chatter in multiple configurations without manual input. The only manual task is classifying the training data. The learning process, and classification is done automatically without any manual thresholding. Furthermore, the trained system is accurate in the testing conditions.

It was also demonstrated that there was little entropy in the audio data, since it compresses down in dimensions easily. While the chatter and not chatter data were close together, this might be an artifact of compressing down to only two dimensions. The compressed data had an intuitive visualization, which is encouraging for expanding this technique to identify more than just chatter.

Conclusion

Previous methods for detecting chatter used manual methods to extract chatter. This was either using a linear transform and filtering out the unneeded signals or extracting features in the data and applying a non-linear mapping to it. None of the methods have applied machine learning to the data without manual manipulation. Machine learning has been used in other areas of research but has not been applied to chatter detection. The machine learning techniques proposed in this project are non-linear extensions of the methods used for standard chatter detection. The autoencoder transformed the data into a lower dimensional space, in which chatter and non-chatter data were distinctly separated. This made the approach effective and allowed for no manual data manipulation.

For the future

While the results look promising, they may not be generalizable. There was a limited amount of test cases, and machine learning requires a plethora of data. The autoencoder could have just selected an encoding where certain tests are close together. More data should be collected from a wider variety of tests for a more conclusive result.

There are several other approaches that should be attempted. One downside of the approach in this project is that it is milling machine dependent. The resonance frequencies that this machine has are different from other machines. The autoencoding is learning the specific frequencies in this machine and endmill configuration, and not the characteristics of chatter. A machinist can easily identify chatter on any machine, regardless if they have used it before or not. Thus, future attempts should address generalize the method to be machine independent. One method would be to not use frequency data, but to extract some other features from the signal. The features could be average noise, spectral energy, ratio of high to low frequencies, and so on. These features would not be specific to a certain frequency band; thus, they would generalize between machines. Another approach would be to use frequency data, but to look for certain patterns. This would be similar to speech recognition – many different people can say the same word, but we can still recognize the pattern of that word. This would involve breaking up the frequency domain into smaller bands and looking at patterns in those bands. This would be similar to how image recognition breaks up an image into smaller chunks so that objects can be recognized regardless of where they are in the image, and regardless to the resolution of the image.

Constraint based toolpath planning

Current state of affairs

Industry

CAM software is used to program CNC machines. 3D models are loaded into the software, and a programmer selects tool path strategies to use. Most tool path strategies are straightforward - parallel toolpaths make the tool move in parallel lines, and contour toolpaths follow the contour of the part. These toolpaths are simple to implement, and do not guarantee constant cutting force or even tool engagement angle. Having occasional spikes in the cutting force leads to the tool and machine resonating and creating bad parts. In some cases, if the increase in the force is dramatic enough, the tool can break. In addition, straightforward paths are not time optimal for removing material in complex shapes. To cut all the material as fast as possible, the tool path must be always cutting as much as possible. This means maintaining a high cutting force.

More sophisticated CAM software will include their own versions of 'adaptive clearing' tool path strategies. Adaptive clearing will generate curvy paths that try to keep the tool from experiencing sudden changes in forces. Doing this will keep the cutting forces constant, which allows the tool to cut as much as possible.

Unfortunately, the algorithm behind the toolpath generation is proprietary, and it is not explained what or how the toolpath is generating. A programmer using such toolpaths has to trust the program to generate a curvy looking path that will perform better than a simple toolpath. In addition, the algorithm still requires parameters such as cutting speed and width of cut.

Research

Research for generating tool paths generally fall into two categories - trajectory planning and time planning. Trajectory planning focuses on creating a spatial curve which represents the path that the endmill will follow. This path is only defined in terms of spatial coordinates, and not time. On the other hand, time planning takes a spatial path, and determines when the endmill will be at each position. Both are required to create a toolpath for the CNC. A CNC machine has limits on its acceleration. This means the machine has to accelerate up to speed, and it cannot do infinitely sharp corners. Trajectory and time planning require the path to be smooth and not have jerks and kinks in them.

Typically, tool path generation is done separately - first the trajectory is created, and then the timing is generated for that toolpath. There has been much research in tool path generation, specifically trajectory planning. The goal is to remove all of the material from the stock to create a part. Generating a toolpath is trivial, generating an efficient toolpath is difficult. Ideally, the endmill will be cutting under constant conditions - the forces on the endmill should be constant. This is easy to do if the path is straight, but to create any sort of shapes, the toolpath has to be non-linear. When the path changes direction, the cutting conditions change. For example, the engagement angle - which is the angle of the endmill that is in contact with material - changes as the toolpath curves around a corner. This is shown in Figure 14 below.

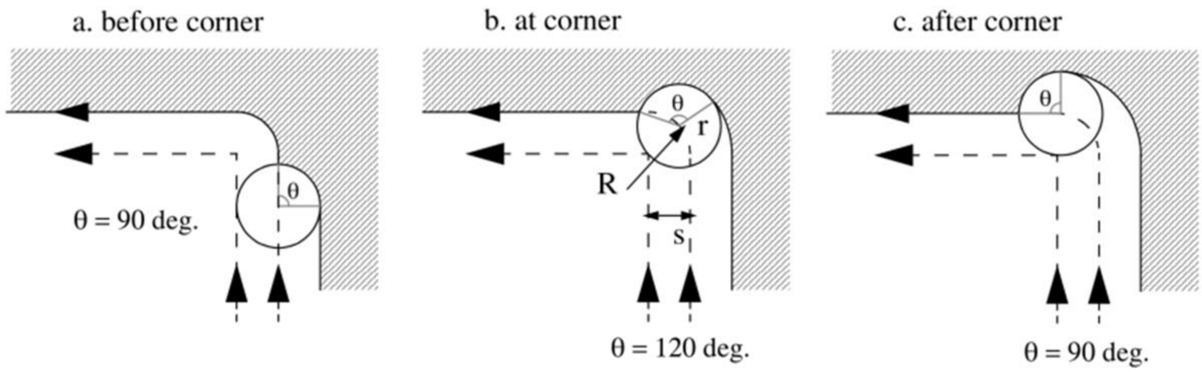


FIGURE 14 TOOL ENGAGEMENT DURING CORNER [9]

The cutting force depends on the engagement angle and the feedrate. In spatial paths, there is no knowledge of the velocity of the endmill, thus engagement angle is used instead of cutting force. In the corner the tool experiences much higher cutting forces. The cutting force is usually the limiting factor for making parts fast. Thus, if a tool path has occasional corners where the cutting force is high, these corners dictate the feed rate and the rest of the tool path becomes overly conservative.

Several different approaches to this problem have been shown in literature. Choy [10] suggests cutting the corners in progressive passes. The tool will pass the corner, but instead of following a parallel path, it will deviate to keep the engagement low. Then the tool comes back and does another pass in the corner. Such a strategy is effective in removing high cutting forces in parallel contour paths. Unfortunately, the toolpath can only be as effective as a parallel toolpath in non-corner segment. Parts that are curvy and that lack distinct corners cannot be machined efficiently with this method. Furthermore, this strategy is only useful for inside pockets.

Another approach, proposed by Hongcheng Wang [9], optimized a contour parallel tool path. The parallel tool path is converted to a spline, and the spline control points are optimized progressively. The cost function is a combination of the endmill engagement angle and curvature of that path. This tries to keep the path from having too high engagement angle and keeps the endmill from making sharp turns.

Overall, the method is effective at removing the spikes in engagement angle. However, its biggest improvement is removing sharp corners from the parallel contour tool path. Sharp corners are inefficient since it means the machine must decelerate and accelerate, wasting time. However, the approach still uses the contour parallel tool path as a basis, and tool path segments with high cutting forces still remain. The path is only modified and is still limited by the initial contour tool path.

While the previous approaches suggested improvements or enhancements of a known toolpath, there have been several different approaches in creating a new tool path generating strategy. These strategies involve using a heuristic to generate a path. Wang [11] suggests making the endmill follow a trochoidal movement. A trochoidal movement is made by a path rotating around a moving center. The radius of the rotating can vary as well. The center of the trochoidal path follows a path that is equidistant from both walls in the part. The radius of the trochoidal path is such that the endmill touches both walls. This is shown in Figure 15 below.

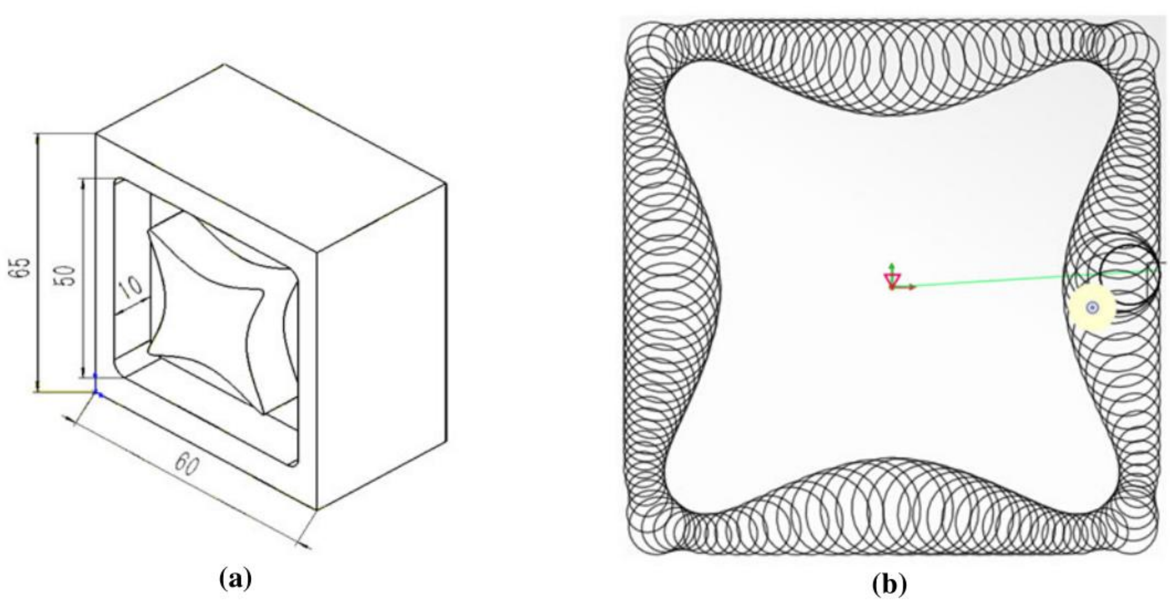


FIGURE 15 ADAPTIVE TROCHOIDAL TOOLPATH [11]

In this manner, the trochoidal path can mill out the part in smooth continuous motions. Furthermore, the trochoidal nature of the path ensures that the engagement angle never exceeds a certain limit, and changes smoothly. This strategy is efficient for high speed machining of pockets. While this strategy can keep engagement low and smooth, it is also inefficient. The path makes the tool make circles, but the tool is only cutting for part of the circle. Furthermore, the trochoidal path only makes the tool reach maximum tool engagement at one point during a rotation. If the pocket is wide enough, this becomes inefficient. Since the engagement angle is lower than the maximum possible engagement, the tool is being underutilized.

Tool engagement control by Dumitrache

Dumitrache [12] proposed a strategy to create a tool path that controls the engagement angle. Since the framework of this approach is the bases for the toolpath generator in this thesis, it is beneficial to describe his method in detail.

The method uses a discrete approach to modelling the stock, part, tool path, and the cutting process. The part and stock are represented as 2D images, or a depth map. Each pixel represents the height of material at that point. In the case of 2.5D milling, all the cutting happens at the same height, thus the height information is unnecessary. The image can be simplified, and each pixel can represent if there is material at that pixel or not.

The algorithm to generate the tool path uses discrete steps. During each step, the next position of the endmill is selected. The selection of the new position only depends on the current situation – the stock remaining, the position of the endmill, the direction of the endmill, and the engagement angle.

The algorithm uses a state machine to determine what action to take. Initially, the algorithm needs to find a starting point. This is either the stock pixel farthest away from the part, or a transition pixel (this will be elaborated later). Once a starting point is found, the algorithm switches into either constant engagement state or contouring state. In constant engagement, a direction is selected to try to get the engagement of the endmill to the desired goal. However, sometimes this direction will cause the endmill to cut into the part. In this case, the state machine switches to contouring state. In this state, the direction is picked such that the endmill touches the part (follows the parts contour) but does not cut into it. If the engagement angle while contouring exceeds the goal engagement plus a certain margin, the state is switched into constant engagement mode. These state changing rules are visually summarized in Figure 16 below.

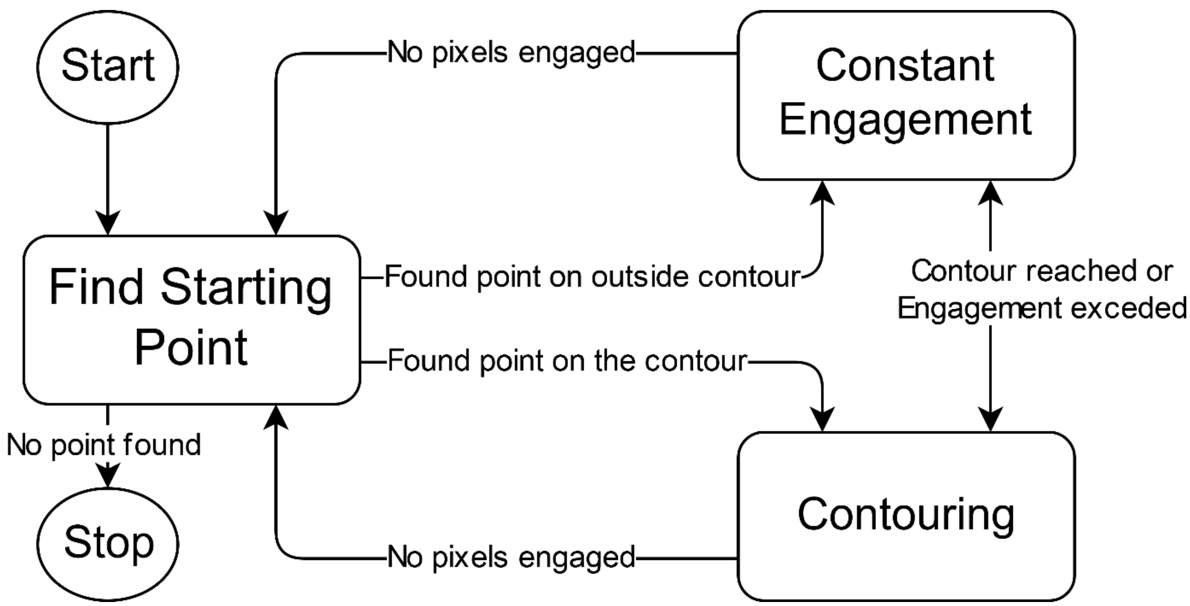


Figure 16 State change diagram

During the constant engagement state the algorithm picks the next direction for the endmill such that the engagement angle goes towards the goal angle. This is best explained with an example and diagram shown in Figure 17. For example, if the endmill is moving to the right, and the current engagement angle is 45 degrees. The set angle is 60 degrees, thus the error is 15 degrees. The direction of travel of the endmill is changed by 15 degrees.

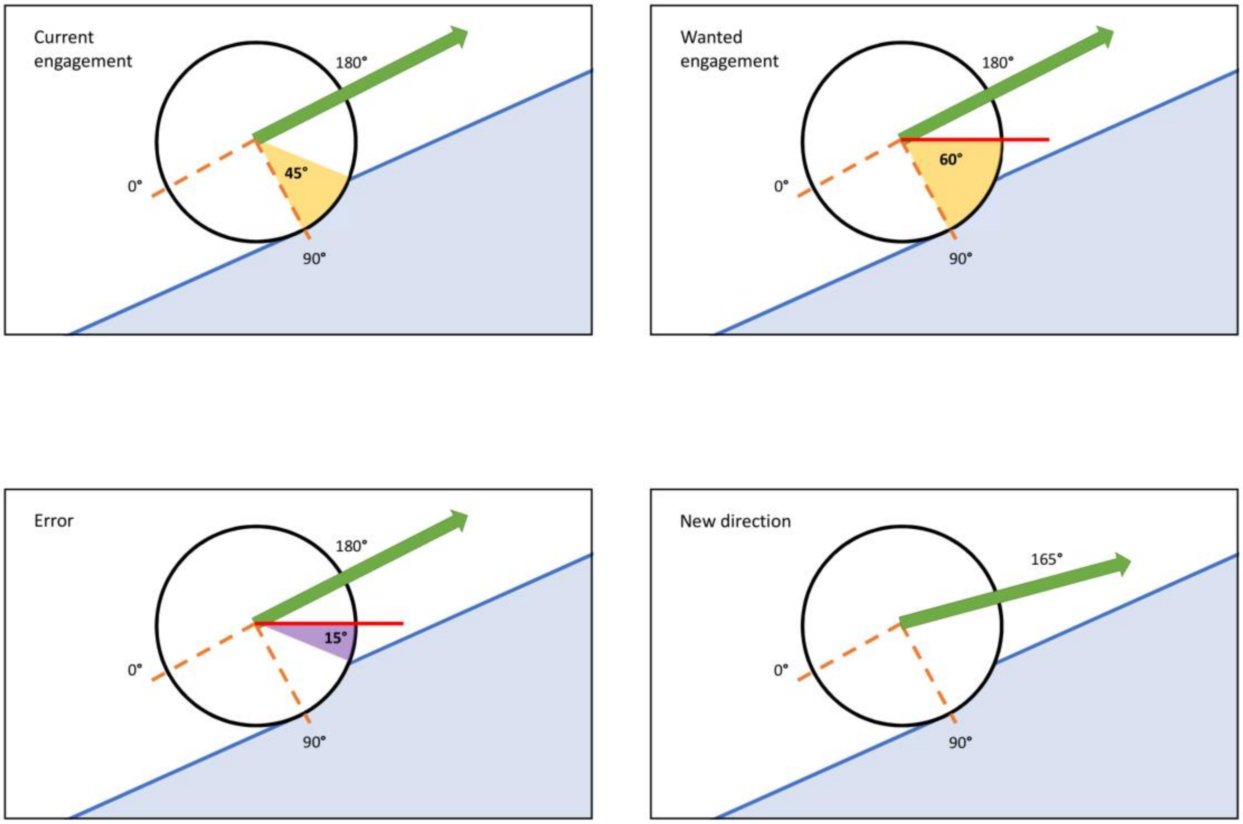


FIGURE 17 EXAMPLE CASE FOR TOOLPATH ALGORITHM

During each step, 180 degrees is defined as the direction of motion. From there, 0 degrees is the back of the endmill, and 90 degrees is where the material makes contact. The endmill will be touching the material between 90 and 270 degrees. The tool cannot be engaging more than 180 degrees of material because it is moving forwards into the material.

The contouring state is entered when the endmill touches the part. During this state, the direction is chosen to be parallel with the part. This make the endmill contour the part.

The algorithm is effective at creating a toolpath without having too many spikes in engagement. Furthermore, the approach is useful for any type of toolpath and part geometry, and is not limited to just pockets or islands. The process of generating the toolpath is not computationally intensive either.

The heuristic used for picking the next direction generally keeps the path smooth but will have sharp corners if there are sharp corners in the stock.

Nonetheless, the tool path has some downsides. The algorithm has a threshold for switching between the cutting states. The threshold allows for the engagement to exceed the set engagement angle. Furthermore, while the heuristic is generally correct, in some cases there will still be spikes in engagement. For example, if the endmill is approaching a sharp angle, the algorithm does not use this information. It will move forward until there is an error in the engagement angle. Thus, for endmill direction to change, there must be an error in the engagement angle.

Project definition

The focus of this project will be to develop improved engagement toolpath stategy combining techniques used in the previously discussed papers. Optimization techniques help select a solution. This solution is not necessarily optimal globally (the best possible solution) but may be locally optimal (better than similar toolpaths). A globally optimization of an entire path at once is infeasible, and is limiting. Changing the beginning of the path will influence the engagement angle of the later parts of the tool path. Thus, a technique that generates the tool path in segments or iteratively would be more suitable and effective.

This project will try to achieve the following improvements compared to past papers:

- Lower engagement angle spikes
- Ability to specify constraints

Lower engagement angle spikes will mean that the endmill can traverse the path faster, and thus more efficiently. Furthermore, constraints can be used to make toolpaths more feasible. For example, a constraint on the maximum turning radius can be added based on how fast the machine can accelerate and decelerate.

Theory

Optimization is finding a set of values that best fit certain conditions. Typically, the goal is to find some inputs to a function that maximizes or minimizes the output of the function. There are many variants of optimization, but the relevant one is numerical optimization. In such a case, the function is defined numerically, and can be computed at any input. Derivatives and other properties of this function are not known. An optimization algorithm will iteratively evaluate various inputs to go towards a possible solution.

Functions that are being optimized can have single inputs, or multiple inputs. A single input function is straightforward to optimize. A function with multiple inputs is much harder to optimize. Changing one input may drastically change how the other inputs affect the output.

Optimization problems can also have constraints. Usually these are either bounds on inputs, or they are constraint functions. A constraint function takes the same inputs as the function to be optimized, and its output has to be greater than, or less than a given constant. This means that some inputs should not be considered for the optimization.

Implementation

This tool path generator will take in 2D images representing the part and stock, and it will generate a toolpath. Initially, the toolpath generation will be based on a heuristic, based on the works of Dumitrache. The next steps will be to adapt the iterative process to include optimization. All of this is built upon a framework that models the cutting process. To summarize, this project is divided into three parts:

- Framework - modelling the cutting process
- Heuristic path generation
- Optimization based

Frame work

The first step to implementing the tool path generator is to have the framework for modeling the cutting process. This involves keeping track of where the tool is, what material has been removed, what material needs to be removed, and the engagement angle of the tool. A 2D pixel based model is used for its simplicity and versatility. This involves having a 2D image that represents the stock, and another image that represents the part. The endmill is a discretized circle, and can only be positioned on pixels, and not in between pixels. While this limits accuracy to the pixel size, the approach is generic and it is easy to model any sort of toolpath and part geometry. Furthermore, this allows for the iteration of the tool path generation to be discreet. The endmill will move a certain amount of pixels in each iteration. The pixels removed can be calculated by taking the intersection of the endmill pixels and the stock pixels. Keeping track of the tool path is also straightforward - the path is just a collection of coordinates that the endmill has visited. In a parametric approach, every curve would have to be stored as an equation. The toolpath would be a collection of equations. While this might work for simple shapes, this will introduce needless complexity if part shape is not simple.

The framework also keeps track of the following properties:

TABLE 3 PROPERTIES OF FRAMEWORK

Property	**Description**
Part image	An image that describes the part. It represents a top down view of the

	part. A black pixel is material to be left, and a white pixel is material to remove. This property stays constant.
Stock image	Similar to the part image, but describes the stock remaining. The changes to reflect the current state of cutting.
Stock contour	The contour pixels of the stock. This is used to find starting locations for the endmill.
Part distance field	Distance field of the part image. Used as a constraint so that the endmill does not cut into the part. Also used for calculating the farthest point from the part, which is a good heuristic for a starting point. Distance field calculates the distance from each pixel to the closest 'part' pixel.
Material left	The amount of pixels left that represent uncut material. This is the difference between the part and the stock image.
Set engagement angle	The set engagement angle that is desired.
Endmill image	A top down image of the endmill. This is a discretized circle. The image is shifted and compared with the stock image. Overlapping pixels are considered cut pixels.

The framework also has the following functions:

TABLE 4 FUNCTIONS OF FRAMEWORK

Function	Description
Mask and image select	Compares a given image to a value, and returns all of the pixels in a given mask. For example, the stock image is given, with the shifted endmill as the mask - all of the cut pixels are returned.
Get angles	Takes a cut pixel, the center of the endmill, and the direction of travel of the endmill. Returns the angle of the cut pixel from the center, relative to the direction of motion. Used for calculating the engagement angle.
Cut stock	Cuts the stock image given the position of the endmill, and direction of travel. Uses the mask and image select function to remove the pixels, and updates all of the properties as needed. Returns the engagement angle of the cut. Can also perform the same operation without removing the pixels, which is used in the optimization function.

Table 4 gives the given functions used to implement various toolpath algorithms. Cut stock is the function used by the toolpath generation algorithm to update the process. Keeping track of the tool position and picking the next direction of travel is up to the algorithm.

Heuristic toolpath generation

Initially, Dumitrache's approach was implemented. The method was used as a baseline for comparing algorithms and their results. This toolpath and subsequent iterations were tested on the test part shown in Figure 18.

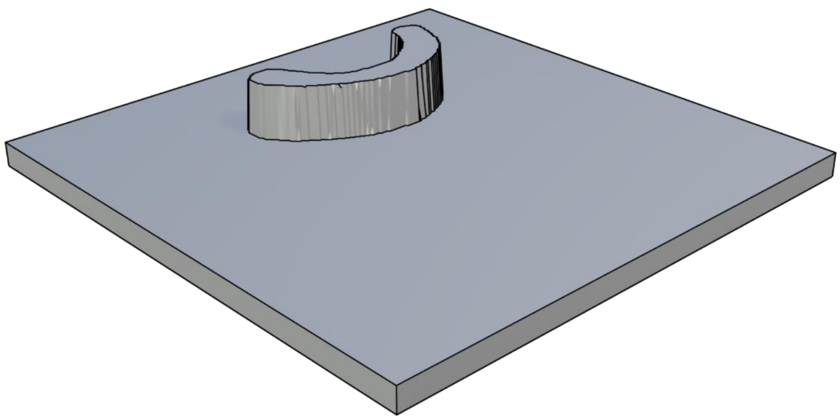


FIGURE 18 SAMPLE PART

Figure 19 shows the results given for the 'U'-shaped test part, and a set engagement angle of 90 degrees.

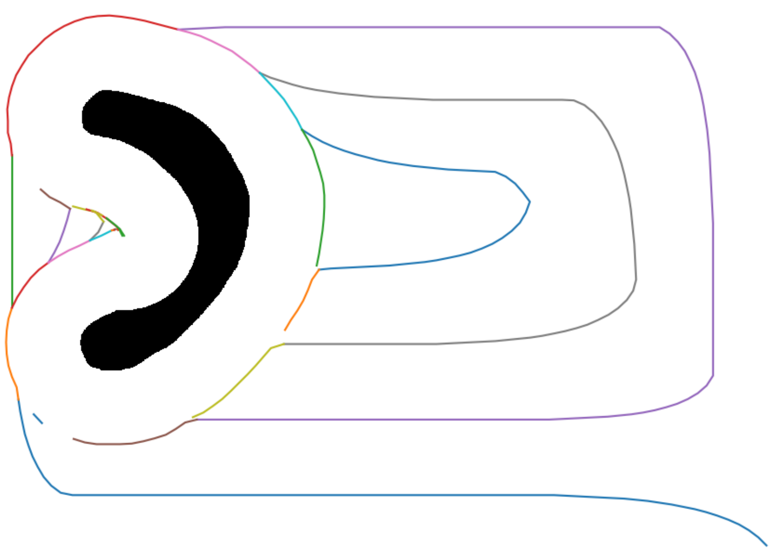


FIGURE 19 DUMITRACHE'S APPROACH TOOLPATH

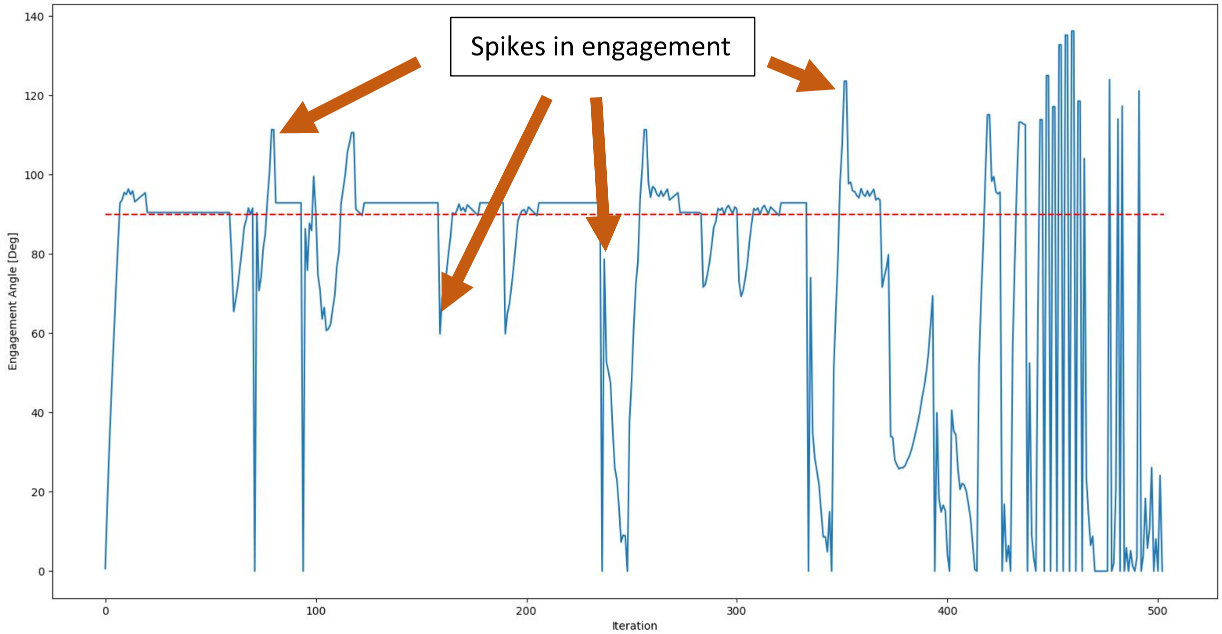


FIGURE 20 THE ENGAGEMENT ANGLE FOR THE TOOLPATH USING DUMITRACHE'S METHOD

The toolpath generates mostly smooth toolpaths. The engagement angle does stay close to the set angle, but during certain parts of the path the engagement spikes up or down. Rapidly changing the engagement angle would mean the tool experiences a high change in force. Furthermore, when the toolpath is doing contouring, it is allowed to increase the engagement angle higher than the set threshold.

The biggest issue with Dumitrache's method is having states. Having a threshold to switch from states is non-optimal. The hysteresis for switching states was used so that small changes in the engagement angle would not keep switching between the states. However, this can be solved with a stateless approach described below in Figure 21.

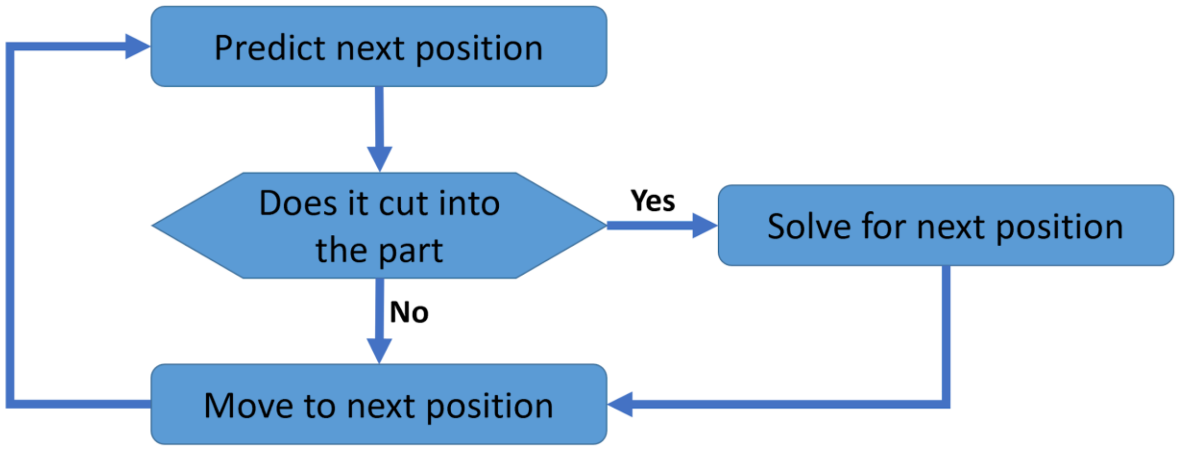


FIGURE 21 FLOW CHART OF STATES

Instead of having separate states for contouring, the algorithm will change the direction of cutting so that it will not be cutting into the part. This is simply done by checking directions that are progressively farther away from the part. One of the angles will move the endmill such that it no longer intersects with the part, and that direction is chosen. The results from this approach are shown in Figure 22 and Figure 23.

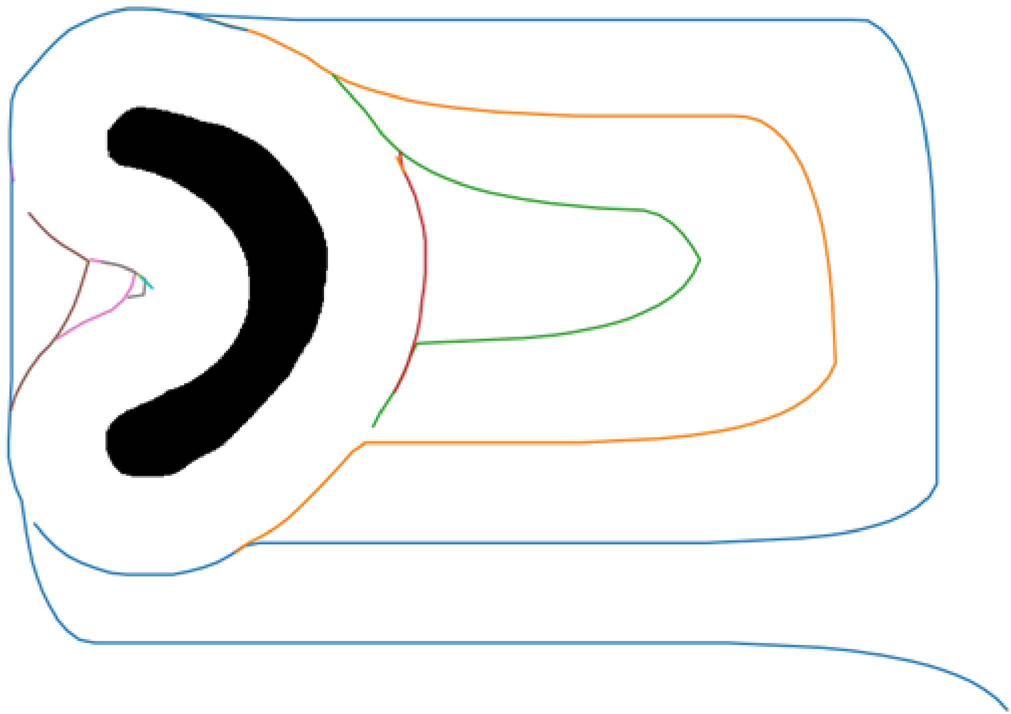


FIGURE 22 NO HYSTERESIS TOOLPATH

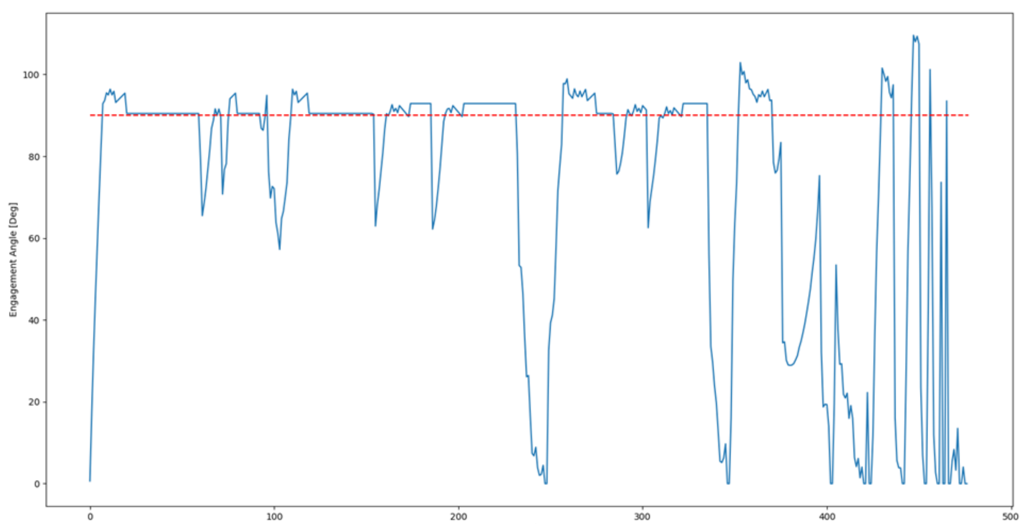


FIGURE 23 NO HYSTERESIS ENGAGEMENT

The engagement angles are lower compared to the previous method. However, the benefit of this approach is not fully utilized. Not having states, and only have one generic approach to generating a path means that it easy to apply optimization techniques to it.

Optimization toolpath generation

Since the toolpath generation is now generic and does not have states it can be put into an optimizer. Each step involves making a prediction, and then executing on this prediction. The prediction might not be optimal, so the prediction is used as the initial guess in the optimization function. Locally, the engagement angle behaves in a nice way - as the endmill moves away from the material, there is less engagement, as the endmill moves towards the material, there is more engagement. Since the initial guess is fairly good and the cost function is well behaved, the optimization should easily converge and yield good results. The approach can be summarized in the following flow chart:

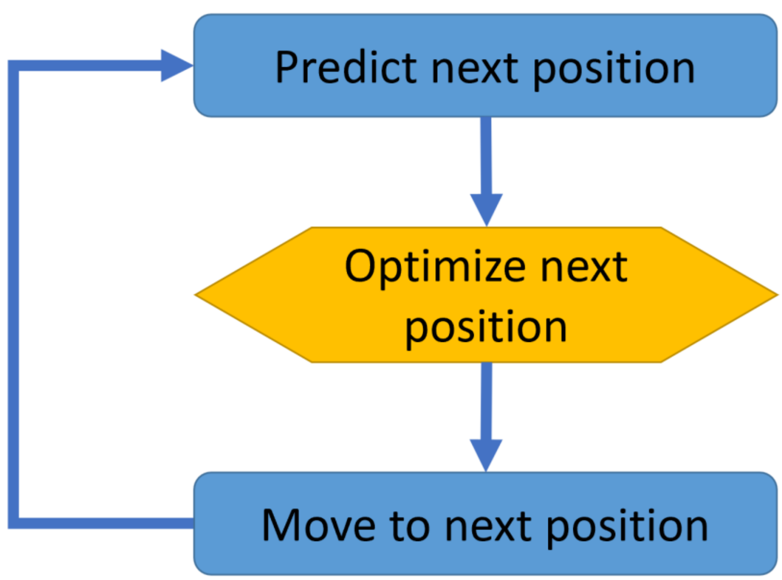


FIGURE 24 OPTIMIZED STATE FLOW CHART

The optimizer takes in a function that outputs a cost, and it also accepts constraints. In this case, the function to optimize is the 'cut stock' function. It takes a direction, which is the value being optimized, and outputs an engagement. The input is a one-dimensional value - the angle of the direction. The error between the engagement the set engagement is the error and is used as the cost.

This optimization will try to keep the engagement constant but is unaware of the part. This can be handled as a special case - if the optimizer cuts into the part, solve for a solution that avoids the part. However, introducing special cases leads to edge cases, and makes it harder to keep the solution optimal. Not cutting the part can be considered as a constraint in the optimization. The endmill must never touch the part. The distance between the endmill center and the closest part pixel should always be greater than the radius of the tool. The distance from each pixel to the closest part pixel is a distance field. This property is precalculated once, since the part does not change. Thus, the constraint simply becomes that the distance field value of the current endmill position is greater than the endmill cutting radius. This is a well-behaved constraint as well. If the endmill is near the part - moving towards the part decreases the distance and moving away from the part increases the distance.

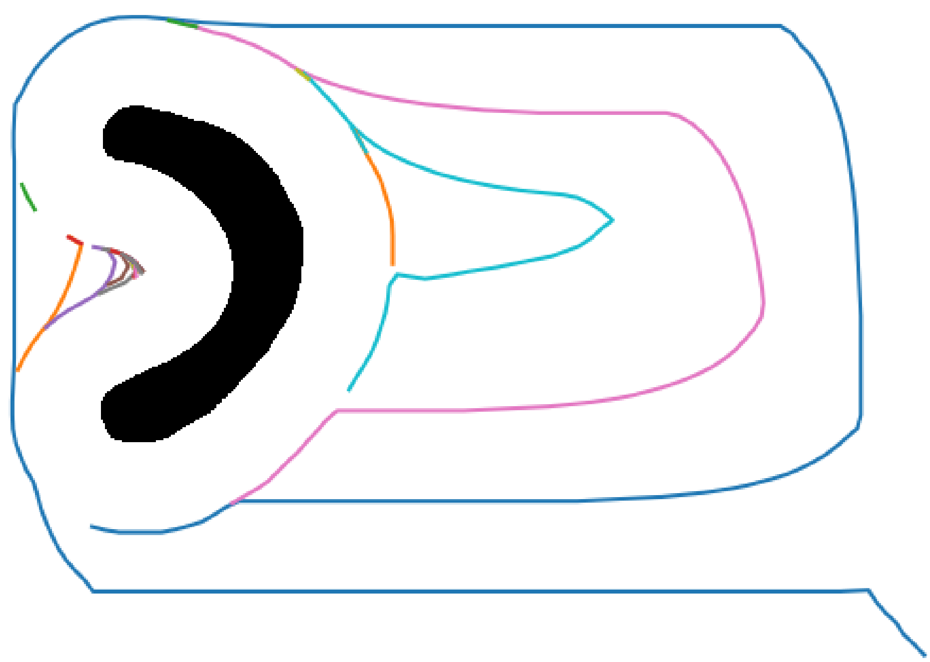


FIGURE 25 OPTIMIZED TOOLPATH

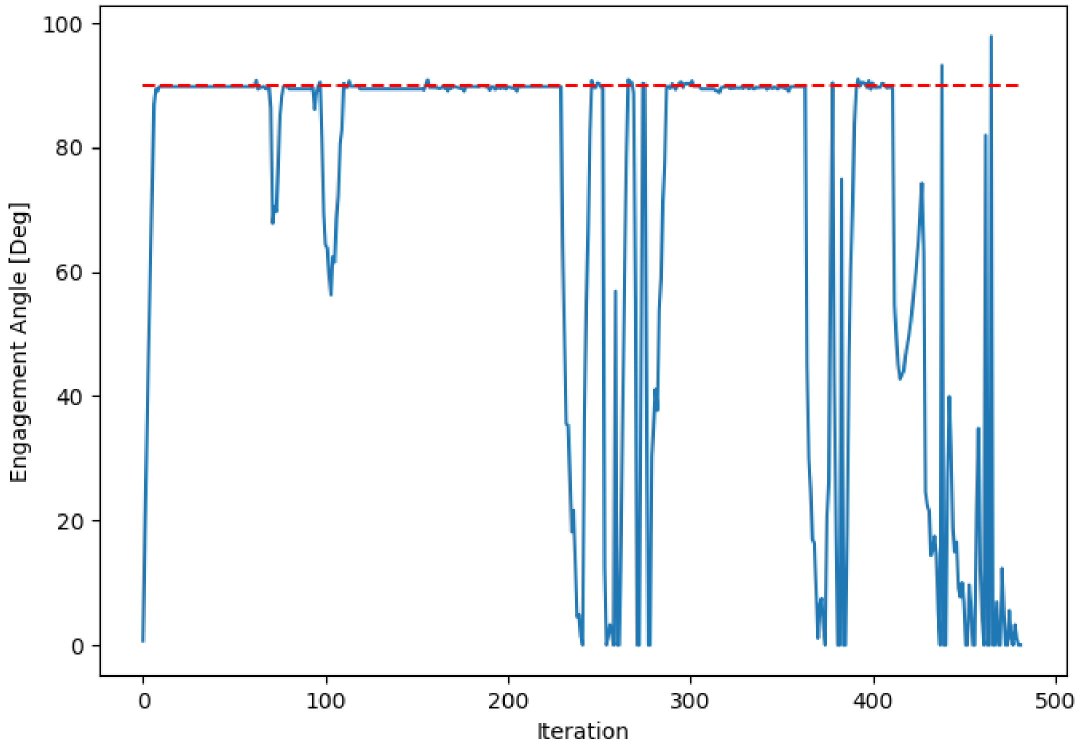


FIGURE 26 OPTIMIZED ENGAGEMENT ANGLE

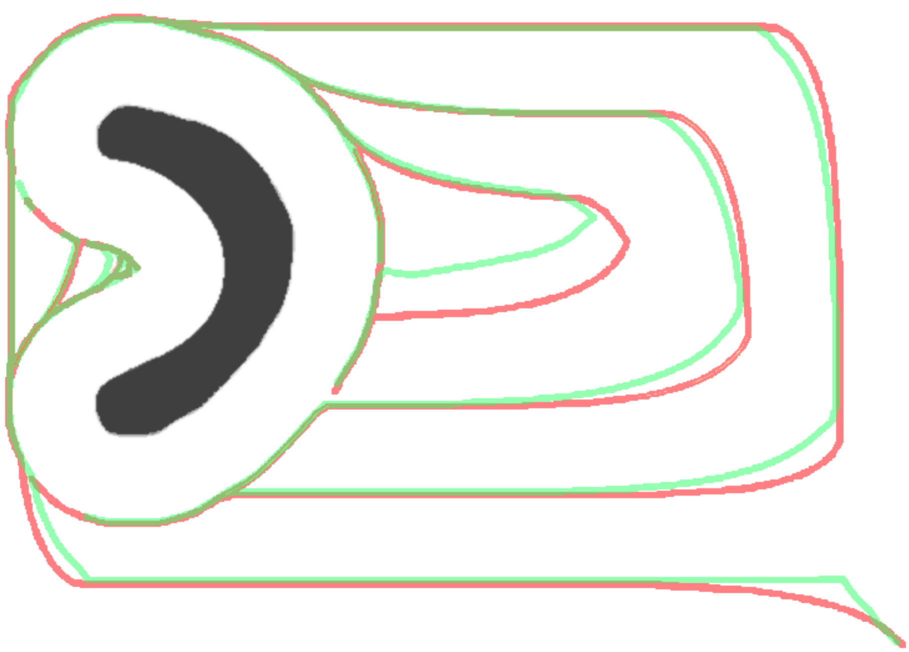


FIGURE 27 COMPARISON OF OPTIMIZED AND NON-OPTIMIZED TOOL PATH

The optimization technique provides good results in terms of engagement angle. The engagement stays close to the set angle, with only a few spikes significantly above the set angle. Those spikes last for only one iteration. This happens when the endmill tries to reach inside the high curvature area of the part. A zoomed in plot of the engagement angles is shown in Figure 28 below.

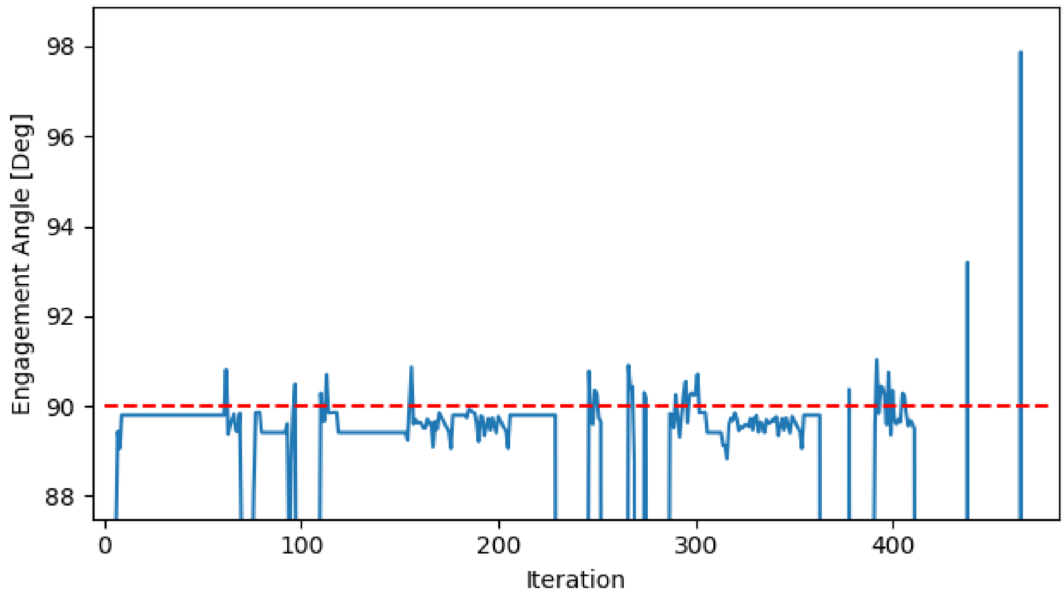


FIGURE 28 OPTIMIZED ENGAGEMENT ANGLE

While the engagement angle results are good, there are some downsides to the method. First, the optimization is greedy. It will pick the closest angle to the set point regardless if that step will allow for good steps after it. Purely from an engagement point of view, the toolpath is close to a theoretical constant engagement toolpath. However, this curve is not smooth. When the endmill is cutting along a smooth section of stock, the toolpath should be smooth, and should retain its constant engagement. When there is a corner in the stock edge, there will be a corner in the tool path. If constant engagement is desired, then corners in the tool path are unavoidable. The difference between the toolpaths can be seen in Figure 27.

This is where the constraints become useful. If the desire is to keep the engagement constant, while not making any sharp turns in the tool path, then this can be described as a constraint. The tool path should not make any turns sharper than a certain radius. The toolpath moves in discrete steps, so this has to be converted to an angle that cannot be exceeded.

Figure 29 and Figure 30 show the results for this added constraint on the toolpath generation. Where the engagement angle greatly exceeds the set angle is labeled with a green star.

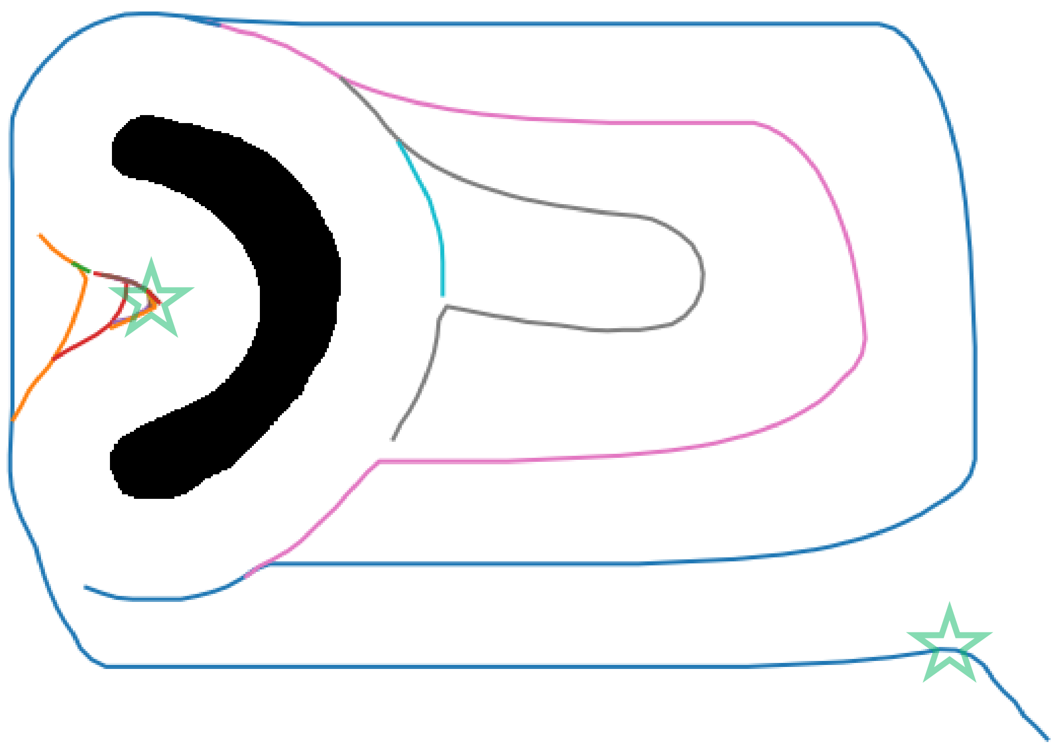


FIGURE 29 CONSTRAINT TOOLPATH

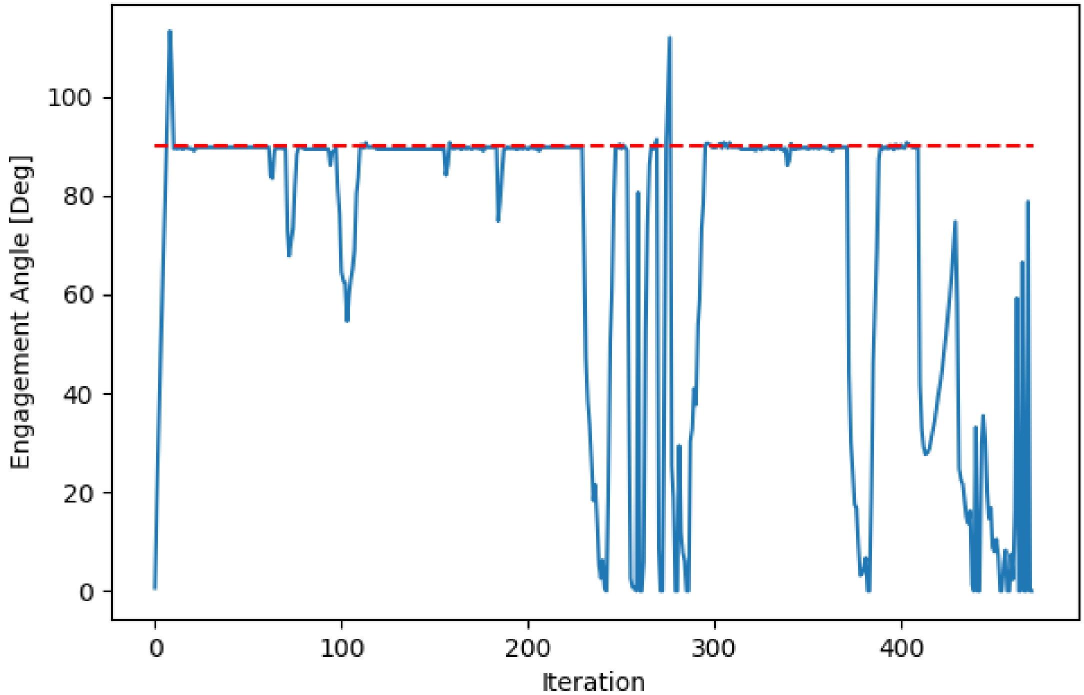


FIGURE 30 CONSTRAINT ENGAGEMENT

When there is a constraint on the smoothness of the path, the engagement angle has locations where it spikes up. There are two notable times when the engagement spikes up. The first is in the beginning - the toolpath is moving into the part, and when the engagement get close to the set angle, the toolpath wants to make a sharp turn. However, the only option is to make a smooth turn, which involves cutting more material. The other spike occurs when the endmill is inside the C part (these are labeled with a star in Figure 29), where the engagement can increase quickly because of the high curvature of the part. Both demonstrate the downside of this approach to optimization. It only optimized the current step but cannot change the previous steps. Thus, a greedy previous step can result in a poor next step and increase the cutting engagement. Another downside of optimizing only future steps is when the endmill reaches the part and start contouring. The endmill could be approaching the part perpendicular path, and when it reaches the part it has two unsolvable constraints: Either crash into the part or make a sharp turn. Since both constraints are mutually exclusive and there is no solution, the backup is to solve for the smallest direction change that does not crash into the part.

Nonetheless, the results are good. Below is a zoomed in plot of the engagement angles. Aside from the two instances of increased engagement, the toolpath stays close to the set engagement.

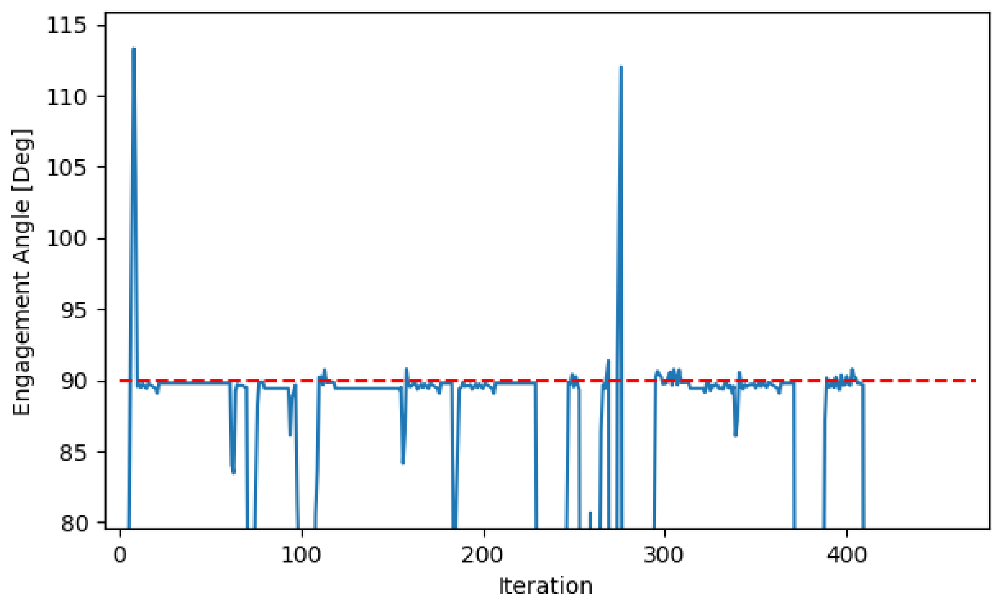


FIGURE 31 CONSTRAINT ENGAGEMENT ANGLE

Multi-step path optimization

The issue with the current single step optimization is that one greedy step can be detrimental to the next steps. An extension of the previous method to resolve this issue would be to look into the future. Instead of optimizing just one step, the optimizer will optimize for multiple steps.

An updated version of the cut stock function has to be made - one that takes in multiple directions. The cut stock function cuts the stock for each consecutive direction and returns the engagement angle for each step. The problem with this is that optimization works on a single cost value, but in this case, there are multiple engagements. The engagements are an N-dimensional vector, and there are multiple ways to calculate the length of a vector. These length assigning functions are called norms. The most common family of functions are called p-norms and are defined as follows:

$$norm_p(x) = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$$

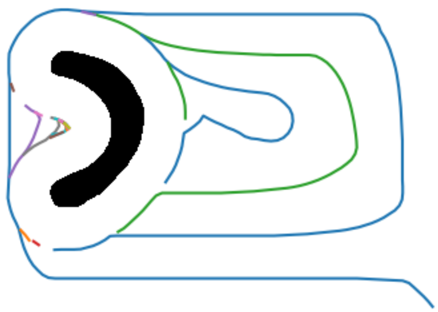
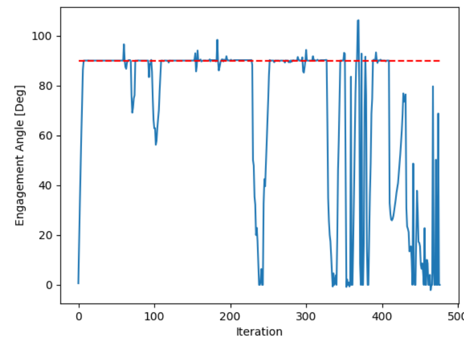
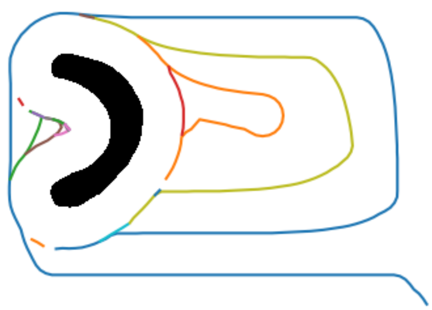
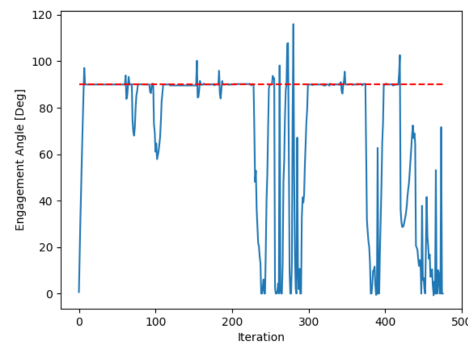
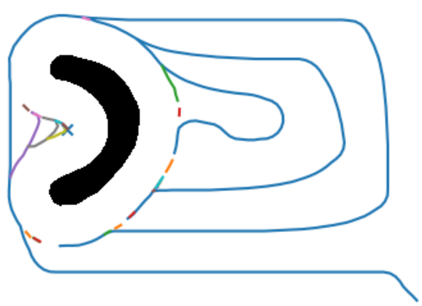
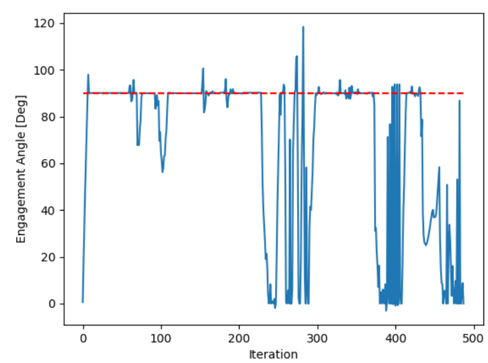
A 2-norm (when p is equal to 2) is the Euclidean distance: all of the distances squared, summed together, and the square of the sum is taken. This is a common metric, but not the only useful one. Below is Table 5 summarizing three common metrics and their properties:

TABLE 5 SAMPLE P-NORMS

P-norm	Description
1-norm	The sum of all of the elements. Each element in the affects the output in the same way, regardless of its value.
2-norm	Euclidean distance. It is intuitive to understand because it is how we perceive distance. The larger the value, the more it affects the output. That is, a small change in a larger value will have more effect on the output.
inf-norm	The distance is the largest value in the vector. Changing smaller value in the vector has no effect on the output. Only the largest value affects the output.

Each norm has different effects on the optimization. With a 1-norm, every engagement will be valued the same, thus, all of them will be changed the same way. With a 2-norm, the step with the largest engagement error will be changed more significantly compared to smaller errors. And with the inf-norm, only the step with the largest error will be optimized. All are valid approaches that produce slightly different tool paths. All three norms were tested, and the results are displayed below in Table 6. Each test used three steps for optimization.

TABLE 6 OPTIMIZATION RESULTS WITH DIFFERENT P-NORMS

Norm	**Path**	**Engagement**
1		
2		
Inf		

Each result had both strengths and weaknesses. For example, the 1-norm did not have any spike when engaging the material in the beginning, but it had several spikes where the optimizer failed to solve and there were sharp turns. The inf-norm was a smoother path (the optimizer could solve in all cases), but had engagement spikes when entering the material.

While the results are promising, using multiple steps slowed down the generation of the toolpath. Even though only 3 steps were used instead of 1, the speed to generate the tool paths was 56 seconds vs 4.5 seconds. This is an order of magnitude slower, while resulting in only a slight improvements to the toolpath.

Results

The toolpath generated using optimization results in tool paths that have near constant engagement. Furthermore, constraints can be added to the optimization to make the toolpath smoother. However, the optimization is not perfect because of its greedy approach. Adding future look ahead to the optimization helps but slows down the generation significantly.

Furthermore, using optimization can allow for feedback in the toolpath. The model and constraints can be updated either during or after the cutting process. Sensors can be used to measure the cutting forces. Once the model is improved, the toolpath can be improved for the next part.

Conclusion

The tool path generator proposed in this thesis tackled a problem with current tool path generators. Dumitrache's approach,, while effective had two major issues. The first is that it did not handle corners well, and the second is that the use of states allowed for spikes in the engagement angle. These problems were addressed by making the approach stateless and by applying optimization techniques to it. This allowed each step generated in the toolpath to be optimal. However, this presented an issue where the most optimal step for engagement was not the best step overall: either it was a greedy choice, or it took sharp turns. This was addressed with two additions: constraints and multi-step optimization. The constraints limited the choices for the next step such that the path was smooth. The multi-step optimization looked at several steps ahead and optimized for the path ahead. Both of these additions allowed for smooth and optimal trajectories.

For the future

While optimization and constraint do make the toolpath generation better, it is still lacking. The model used in this toolpath generation is simplistic. The set engagement angle, and constraints are just guesses. Limiting the turning radius generally helps the CNC to not slow down, however, it is not guaranteed. Maybe it is more efficient to fully stop and make a sharp turn in some cases. If a feed speed is selected for a CNC machine, selecting an engagement angle allows for approximate control of cutting force. However, it would be better to control the engagement angle and feed at once.

For better results, a more realistic simulation is needed, a simulation where the cutting force and time are considered. A similar optimization approach can be used, but instead of taking discrete steps, a discrete time step is taken. The constraints should be physical machine limitations: cutting force, accelerations, etc. This should provide a more optimal toolpath. Instead of selecting an engagement angle, the optimizer will select the best feed and speeds that achieve the fastest material removal.

A problem encountered during this project was the optimization solver itself. It was treated as a black box. However, it might be possible to create an optimization algorithm that is more suitable for this task.

The path optimization used an angle for each step. However, this might not have been the best choice for multiple steps. Since each step is based on the position of the previous step, changing the initial angle changes the position of every other step. This means that if an optimal path of the last steps are found, but the initial step is changed, the entire path has to be reoptimized. Each step in the path is not linearly independent. However, if instead of an angle, each step directly used the position, each step can be optimized independently from each other.

Automatic tap tester

Theory

The previous section showed a toolpath generator that used a cutting model to generate toolpaths. The weakness of the toolpath generator was that it relied on a cutting model. Calculating the cutting forces on an endmill while the cutting conditions do not change is straightforward [3]. However, cutting conditions are never static while cutting. Furthermore, it is not only necessary to know the cutting force, but how the endmill and machine will react to that cutting force. In certain cases the machine will chatter. To be able to predict these phenomena, a dynamic model is needed.

Objects have a certain amount of stiffness. The more force that is applied to them, the more they deform. In the simplest case, this is modeled as a spring. For a spring, the deformation is proportional to the force applied, and the ratio of the deformation to the applied force is the stiffness. However, an object's stiffness is also dependent on the rate of change of the force. The stiffness changes depending on the frequency of the applied force. A spring with a mass on it will have a certain stiffness when slowly pushed. But if the applied force is at the resonant frequency, the spring will be much less stiff - it will take less force to displace it the same amount.

Typically, ‘Frequency Response Functions’ (FRF) are used to describe the frequency dependency of stiffness. FRFs plot the stiffness with respect to the frequency input. To create an FRF, one can input a sinusoidal force, and measure the output displacement. This excites the system at one frequency at a time. A more robust approach would be to apply an impulse to the system and measure the output over time. The time domain data can be transformed into frequency domain which is the FRF.

Tap testing involves providing an impulse to the system (a tap) and measuring the output. Using this, the FRF of the machine can be computed. This is relevant to CNC machines because the cutting process involves the endmill cutting into the material periodically. Each cut is an impulse, and under certain conditions can cause the machine to resonate.

Current state of affairs

Industry

Tap testing is used in various industries for modal analysis, however it lacks widespread adoption in manufacturing industries. Most machine shops are unaware of such tools and techniques. Tap testing equipment and software is dominated by a few companies, and it is not easy to learn how to use it. The most common setup for tap testing requires setting up an accelerometer on the tip of the endmill and hitting it with a special impact hammer that measures the forces during impact. The accelerometer can be mounted on the endmill in various ways, and each method has different drawbacks. Impact hammer come in a variety of weights and with tips with varying hardness. Each having its own drawbacks and use case. The barrier to entry for using this equipment is high. Contractors that provide tap testing services exist. However, they have their own set of problems. Setting up the equipment for each tool is time consuming. The tapping process itself requires considerable effort to make sure that it each tap produces usable data. In addition, all of the sensors are connected with thin cables which carries analog signals. It is easy to bend these cables and distort the analog signal coming in. The entire tap testing process requires finesse from someone experienced. This makes such services unscalable

Research

In the academia world, modal analysis and tap testing is a frequently used tool. However, academia suffers the same problems as the industry. They are locked in with a few companies making the tools and software. However, this is manageable since the use cases are for experiments and research. Taking the time to learn how to use the equipment and doing a single tap test is feasible and normal. Since tap testing is not a bottleneck for academia, there is not much research that goes into making alternatives, or automating the process.

There has been research in producing FRFs of a machine without tap testing. Instead the FRF uses data from the cutting process. Such research has been done by Zaghbani [13]. The proposed approach used an accelerometer mounted on the spindle to measure vibrations while the CNC machine performs cuts. The technique used assumes that the input to the system (the cutting) is white noise, thus it excites all frequencies. This method can more accurately predict the damping coefficient of the system, since a tap test only measures the vibration of the endmill in air. While the method only requires an accelerometer, it is lengthy to perform the necessary test cuts.

Dunwoody [14] designed an automated tap tester for his thesis project. The automated tap tester used a solenoid with a piezo-electric force sensor, a solenoid hammer for actuating the tap, and a capacitive displacement sensor. This setup can automatically capture the FRF of a milling machine with little human intervention.

While Dunwoody's method eliminates many problems that typically plague tap testing, it introduces some problems of its own. The impact hammer and the capacitive sensor are on the same side. This means that the hammer hits the bottom part of the endmill, but the capacitive sensor measures the displacement higher up on the endmill. To reconstruct the FRF, the tip displacement has to be calculated using the measured displacement, which introduces errors. The capacitive sensor is also non-linear. The measurement signal to distance function must be approximated each time. This involves moving the endmill in small increments and measuring the sensor output. This is hard to automate, since it would require the device to tell the machine which positions to move to. This introduces both measurement errors and makes the device less automated. Lastly, the force sensor is still the same sensor used for industry impact hammers. It requires expensive equipment to measure the signal.

Project definition

The goal for this project will be to design and build an automated tap tester. It will explore alternative design choices compared to Dunwoody's automated tap tester. The goals for this project will be to create a device that can potentially be used in industry for tap testing. The goals can be summarized as follows:

- Automated - minimal human interaction should be required to operate the device.
- It should be able to tap test multiple tools of various diameters.
- Measure FRF directly - impact the endmill at the tip, and measure the displacement at the tip.

Implementation

The implementation of the automated tap tester has three main hardware components. They are:

- Actuator - the moving part that hits the endmill.
- Displacement sensor - measures the vibrations of the endmill.
- Force sensor - measures the impact force of the tap.

In addition to these components, there are components that connect these together. This includes a motor controller, analog to digital converter, microcontroller and so on.

Actuator

Typically, the actuator in a tap tester is the operators arm. The person swings the hammer at the endmill, the hammer impacts the endmill and bounces back. The person only provides the initial movement to the hammer and tries to have a minimal effect on the impact and bounce back of the hammer. The impact dynamics are mainly dictated by the stiffness and mass of the hammer. When a hammer impacts the endmill, there is a brief moment when the two objects are touching each other. The system can be viewed as a mass spring damper. The endmill deflects, the hammer tip deflects, and the hammer has a mass. The stiffer and lighter the hammer is, the faster it bounces back. A fast bounce back is closer to an ideal delta Dirac impact, which will excite more frequencies. However, the operator who swings the hammer has limited ability to control how fast it moves. This means that stiff, light hammers transfer little energy into the system resulting in a weak signal output. Due to this, there are a variety of impact hammer weights and tip stiffnesses.

A solenoid hammer was used as the actuator in Dunwoody's thesis. The hammer is limited to only one type of tapping. A solenoid hammer moves when the electro-magnet is powered. The force applied by this magnet is not controlled, so only one type of tap is achieved. The bandwidth of the tap is solely based on the mass and stiffness of the hammer.

However, if the actuator movement was controlled, it would be possible to 'simulate' taps that have different bandwidths. For example, once the actuator senses that it has touched the endmill, it can pull back faster than it would have normally bounced back. This means that the bandwidth of the tap can be controlled. If the actuator is infinitely powerful, it would be possible to imitate any type of tap. Realistically, there would be a trade off between bandwidth of the tap and amplitude of excitation.

To build such a tap tester, a linear actuator is needed. The simplest solution would be to use a motor connected to a ball screw linear stage. However, it is necessary to calculate if a motor can move a linear stage at the necessary speeds. Assumptions about the following have to be made: frequency and amplitude of vibrations, size, shape, and mass of the tap tester. Below are calculations done to determine the feasibility of using linear stage for the tap tester.

Milling machines typically have spindles capable of 10'000 RPM, and given a 4 flute endmill, this means that the cutting frequencies will be at 666 Hz. Thus, the dynamics of the endmill past that frequency should be excited for an accurate FRF. Roughly 1000 Hz of bandwidth should be appropriate. Since the vibrations are sinusoidal, the maximum linear velocity and acceleration of the endmill can be calculated. The tap tester must accelerate faster than the endmill to provide the needed bandwidth.

To do the rest of the calculations, a sample linear stage needs to be selected. LS1210-60-T56.4 from MISUMI is selected because it is lightweight but has enough range of motion for tapping various endmills. MISUMI provides a data sheet with the specification of the linear stage, so it is possible to calculate the combined inertia of the table and ball screw. The calculations are shown below.

$$J_{table} = \left(\frac{lead}{2\pi}\right)^2 * mass_{table}$$

$$J_{total} = J_{table} + J_{screw}$$

Given this inertia, and the given acceleration needed, it is possible to calculate the torque and speed requirements of the motor. The calculations are given below.

$$v_{maxlinear} = A_{amplitue} f_{frequency}$$

$$a_{maxlinear} = v_{linear}$$

$$\omega_{rotational} = \frac{v_{maxlinear}}{lead}$$

$$\alpha_{rotational} = \frac{a_{maxliner}}{lead}$$

$$T_{peak} = J_{total} * \alpha_{rotational}$$

The motor that drives this linear stage would have to have a torque of 4.1 newton meters at a speed of nearly 2000 rpm. Furthermore, such a motor would need to be able to apply that amount of torque instantly. Large stepper motors can provide such torque, but the inductance on such a motor would be limiting. It would take a significant amount of time for the motor to start accelerating. Thus, a ball screw linear stage and motor is not an option.

An alternative to a ball screw linear stage would be a voice coil. Voice coils work in the same manner as brushless motors, except they produce linear motion. Voice coils have a winding on the stator, and permanent magnets on the cone, which is equivalent to a typical motor's rotor. When a current is applied to the coil, it pushes the cone. Voice coils are known for their ability to quickly accelerate [15].

A voice coil is current controlled. The current passes through the coil which has a resistance and inductance. This current creates a magnetic field which interacts with the magnetic field from the permanent magnets in the stator. This creates a force on the cone. Once the cone is moving, it creates a back electromotive force (back EMF) - a voltage proportional to the velocity of the rotor. Thus, a voice coil can be modeled as a resistor, inductor, and a voltage element in series. The cone is a mass that is typically on a linear rail which has little friction; thus friction is neglected for simplification. The equations for modelling a voice coil are presented below.

$$V_{resistance} = IR$$

$$V_{inductance} = L\frac{dI}{dt}$$

$$e_{back\ emf} = K_v s_{speed\ of\ coil}$$

$$F = K_t I = m\dot{s}$$

$$V_{in} = V_r + V_i + e$$

$$V_{in} - K_v s = \frac{m}{K_t}\,\dot{s}R + L\frac{m}{K_t}\,\ddot{s}$$

$$\frac{Lm}{K_t}\ddot{s} + \frac{mR}{K_t}\dot{s} + K_v s - V_{in} = 0$$

The equation of motion is a second order differential equation with respect to the speed of the voice coil. This can be solved using a symbolic solver, or numerical methods. There are several different parameters that affect the performance of a voice coil - mass, motor constant, maximum rated current, and inductance. Several different voice coils were analyzed, and their performance was compared. The differential equations were solved for the motion of each coil with a given initial velocity, and their maximum rated voltage. This simulates a voice coil pulling back as hard as it can. This motion is a virtual tap. The time taken to slow down and come back to the original starting position is the duration of the contact of the coil and endmill. If a voice coil can pull back quickly, that means it can perform a tap that has a wide frequency content (wide bandwidth).

Using this method several voice coils were simulated. Voice coils by 'SUPT motion' were analyzed, since they provided both the data sheet for all the parameters, and the voice coils had a low price for purchasing single units. Figure 32 below is a comparison between the best two coils.

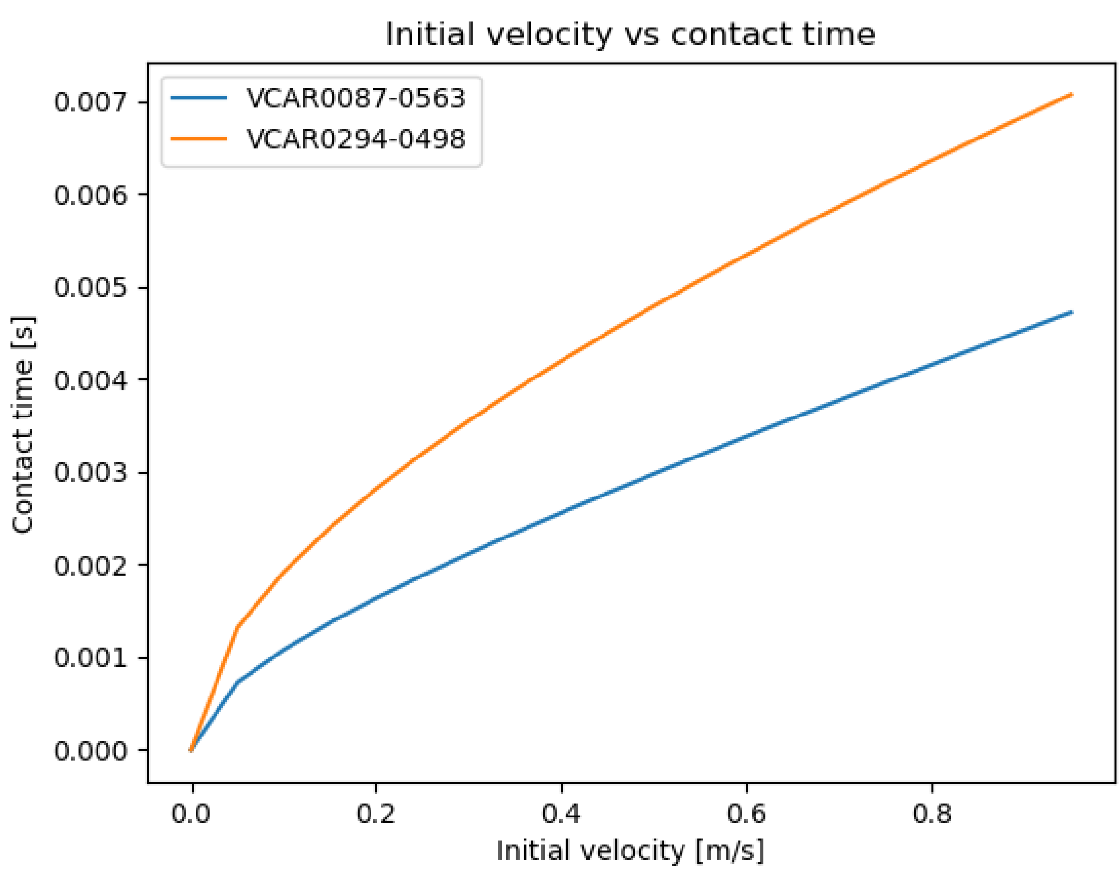


FIGURE 32 VOICE COIL MOTION

The ‘VCAR02294-0498’ voice coil was selected because it had the best performance of all of the voice coils available. At impact speeds between 0.1-0.2 meters per second, the coil would be in contact with the endmill for approximately 1 millisecond. This will provide the necessary bandwidth to excite the system.

The simulated tap only considered the voice coil moving under its own power. However, when the voice coil hits the endmill, the endmill will be deformed and will be pushing back. This should make the tap even faster than the simulation.

Motor driver

Voice coils behave like brushed DC motors in terms of control. Thus, a motor driver suitable for driving brushed DC motors can drive the voice coil. The BasicMirco MCP236 was chosen as the motor driver because it had sufficient current capability and can be easily interfaced with microcontrollers and a PC. However, it became clear that the motor controller does internal processing that causes an output delay. This was tested by filming the motor controller and voice coil in slow motion. A microcontroller sent a command packet to the motor driver, and turned on an LED light. The LED light turned on 20.8 milliseconds before the voice coil started. This means that the voice coil will not be able to be controlled in real time. It is necessary to start moving the voice coil as soon as it touches the endmill.

The MCP236 motor controller has an internal processor that processes the input and then determines the output. While this is useful to do less computation, the computations take too long. Thus, a motor controller with lower latency is required. Unfortunately, manufacturers rarely publish latency numbers of their motor controller. However, it is possible to use a motor driver instead. A motor driver is an H-bridge that chops the voltage. Unlike a motor controller, it does not control the current, so extra consideration has to be taken when using a motor driver.

Unfortunately, it is hard to find a motor driver that supports the appropriate voltage level. Most motor drivers at such levels are either custom built using components or are motor controllers. A compromise was made, and a hobby level motor driver was used. The driver is the Pololu High-Power Motor Driver 36v20 CS. It supports up to 50 Volts input range and delivers up to 20 amps continuous current. The input voltage is lower than the rated voltage of the voice coil, so the downside of using this driver is that it will not be able to accelerate the voice coil as fast. The latency of this driver was tested using a 240fps video recording in the same matter as before, and the latency was below 1 frame (4.2 milliseconds).

The motor driver had one more issue, which is that the ground between the low voltage control inputs and the power ground were connected. The powered output is voltage that is being chopped at a high frequency, introducing noise on the rest of the signals. To isolate the motor driver from the rest of the electronics, opto-isolators were used.

Force sensor

The force sensor is used to measure the impact force. The force sensor is usually decoupled into two parts - the transducer (converts force to an electrical signal), and an analog to digital converter (converts the electrical signal into a digital signal). Different sensors sometimes require different types of amplifiers to condition the signal before it is converted into a digital signal.

Most force sensors do not measure force directly, but actually measure miniscule amounts of deformation. There are several principles that can be used to measure deformation, but the most common are resistive and piezoelectric. Piezoelectric materials produce charge when there is a change in the applied force - the electrons get squeezed out of the crystal when it is deformed. The crystals are durable and have high sensitivity - there is a lot of signal for little force change. However, the crystals create a signal based on the change in force, thus to measure the force, the signal has to be integrated. Charge amplifiers that are used to integrate the signal are expensive. Furthermore, the amplifiers are leaky - if a signal is constant, the charge amplifier will slowly decrease the signal. This means they have to be calibrated before every single tap. In addition, piezoelectric sensors are also expensive, making the entire setup not cost efficient.

Strain gauges are resistive strain sensors. Changes in strain cause a change resistance that can be measured. While the signal from the strain gauges is not sensitive and requires amplification, it is directly proportional to the force. Furthermore, strain gauge-based force sensors - known as load cells - are inexpensive. There are many integrated circuits made to measure such small voltages that come from the strain gauge. This makes the setup both small and cheap. For this reason, a load cell was selected. The sinoder F9604 load cell was selected because it was light and had a reasonable force sensing range.

The other component to be selected was the ADC. There are many ICs that are made for measuring load cell signals. Originally, the AD7124-8 was selected. It had an amplifier, and a 24 bit ADC. However, that ADC has an internal filter with a bandwidth of only 50 Hz. Unfortunately, most ICs for measuring the voltage from load cells have built-in filters at low bandwidths. Ultimately the ADC LTC2500-32 was chosen. The built in filter can be configured to have a bandwidth of 1/4 of the sampling rate. Thus, an adequate sampling rate can be chosen to make sure the bandwidth requirements are satisfied. The ADC has no built in amplifier, however, it has 32 bits of precision, which makes up for the lack of amplification.

Displacement sensor

When the endmill is tapped, it will vibrate. This vibration and its decay needs to be measured to determine the FRF. Traditionally accelerometers are used because of sensitivity to vibrations. Miniscule displacements at high frequencies results in high accelerations. However, the biggest downside to using accelerometers is that they have to be physically attached. Not only does this require a person to manually attach them, it also interferes with the measurement. Small endmills are impossible to tap test because the accelerometers are too large to attach.

For an automated tap tester, it is necessary to use a non-contact sensor since it will not require manual setup. There are several different types of sensors that can measure displacement or vibration without contact. The most common ones are capacitive sensors, eddy current sensors, laser interferometers, and laser displacement sensors.

Capacitive sensors use capacitive coupling to measure the distance to an object. When an object moves towards the sensor, there is a change in the capacitance. This change is measured and correlated to a distance. However, the relationship between the signal and the distance is non-linear and would have to be calibrated. Furthermore, the signal changes based on the material type, since different materials have different conductivity.

Eddy current sensors use a coil to create a magnetic field which interacts with a conductive material. This eddy current creates a magnetic field that can be measured. Just like the capacitive sensor, the object has to be able to conduct electricity to be sensed. Furthermore, the signal is nonlinear compared to the distance of the object and depends on the type of object.

Laser interferometers bounce a wave of light off an object and measure the phase shift in the light. They are extremely precise, and do not depend on the material of the object. They output is linear with respect to the displacement. Unfortunately, they are expensive and require a bulky equipment to operate.

Laser displacement sensors use triangulation to measure distance. A beam of light is bounced off the object, and the angle the position of the returning beam is measured. The signal is linear with respect to the distance, and can measure absolute distances. The measurement requires a certain amount of light bounced back to the sensor for the measurement to be accurate. The sensor is much cheaper and smaller compared than most of the other sensors and can be self contained in a small package.

The downsides of nonlinearity of eddy and capacitive sensors make them non-ideal for use in an automated setup. Laser interferometers are a good choice, but are too bulky and expensive. Thus, the only remaining sensor is the laser displacement sensor. When selecting a sensor, the criteria are resolution and sampling rate. Typical impact forces are in the 500 to 2000 newton range [16]. Carbide endmill with 20 mm diameter will deflect at least 40 micron with a 500 newton impact force [17]. It is necessary to measure the vibration of the endmill, thus the resolution should at least capture 10 points of vibration, thus 4 micron resolution is necessary. The sampling rate has to be at least twice as high as the bandwidth (or 1000 hz) to be able to capture the necessary harmonics. Unfortunately, many laser displacement sensors do not sample at a consistent rate. They have variable shutter times (to allow for variable sensitivity to maximize accuracy of measurement), which means that they cannot be used to measure vibrations.

Opex FA makes a laser displacement that has both an internal clock, 0.25 micron resolution, and 80 kilohertz sampling rate. This sensor was chosen as the displacement measurement sensor.

Electronics selection

All of the components have to be orchestrated to work in tandem to complete a tap. The voice coil has to be controlled in real time with measurements from the force sensor. The laser displacement sensor data does not impact the voice coil, thus, the data can be collected separately and joined together later. An embedded microcontroller is needed to interface with the motor driver and force sensor. The microcontroller can send all of the data back to a computer that is interfaced with the laser directly.

For the microcontroller, the Nucleo-144 STM32F429 was selected. It has a fast processor and ample onboard memory, which makes it easy to program without worrying about performance and memory usage. It has a plethora of IO pins, which makes it easy to connect all of the components.

The force sensor is connected to the ADC, and the ADC communicates to the board using SPI for data transmission, and a few digital pins for timing. The motor controller is controlled with PWM pins, which are connected via the opto-isolator. Figure 33 and Figure 34 below show the pins used to connect the components to the microcontroller.

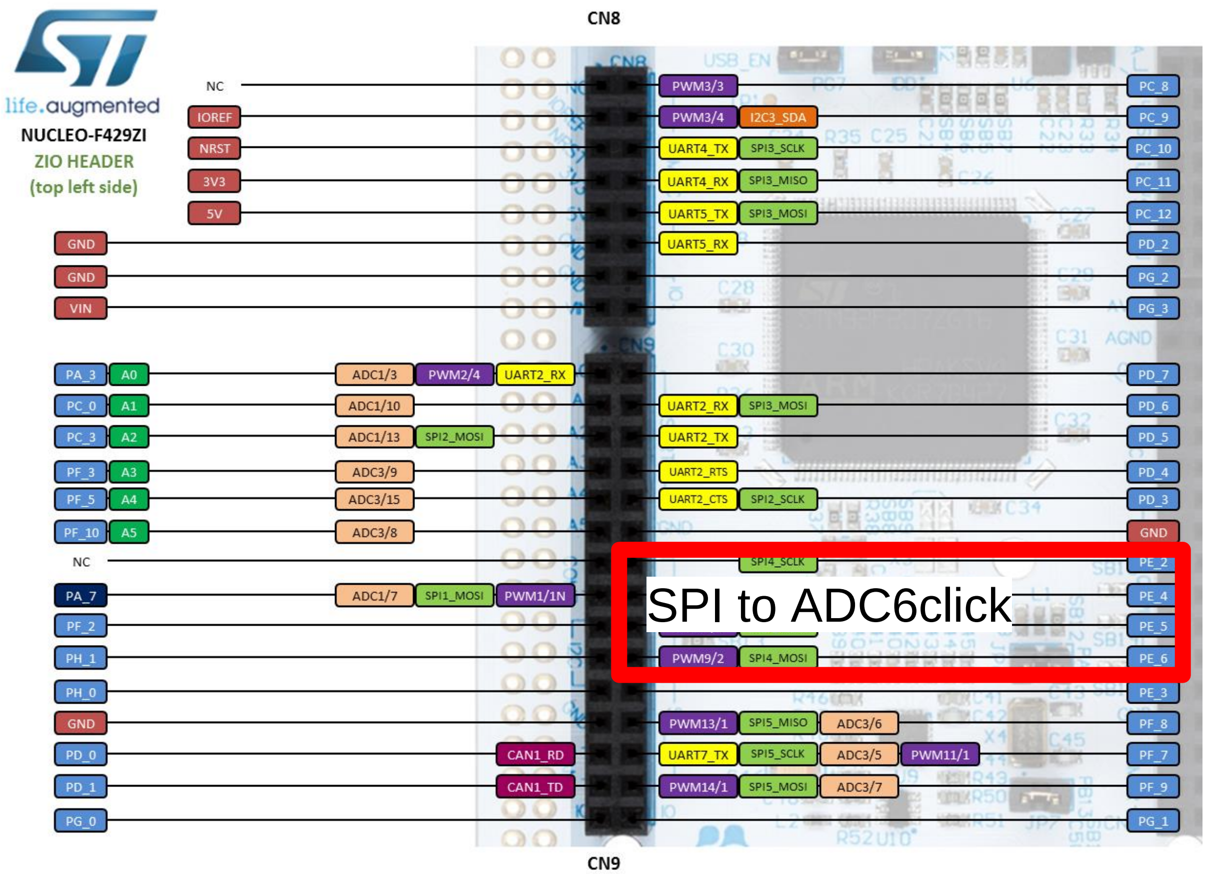


FIGURE 33 MCU PIN CONNECTIONS

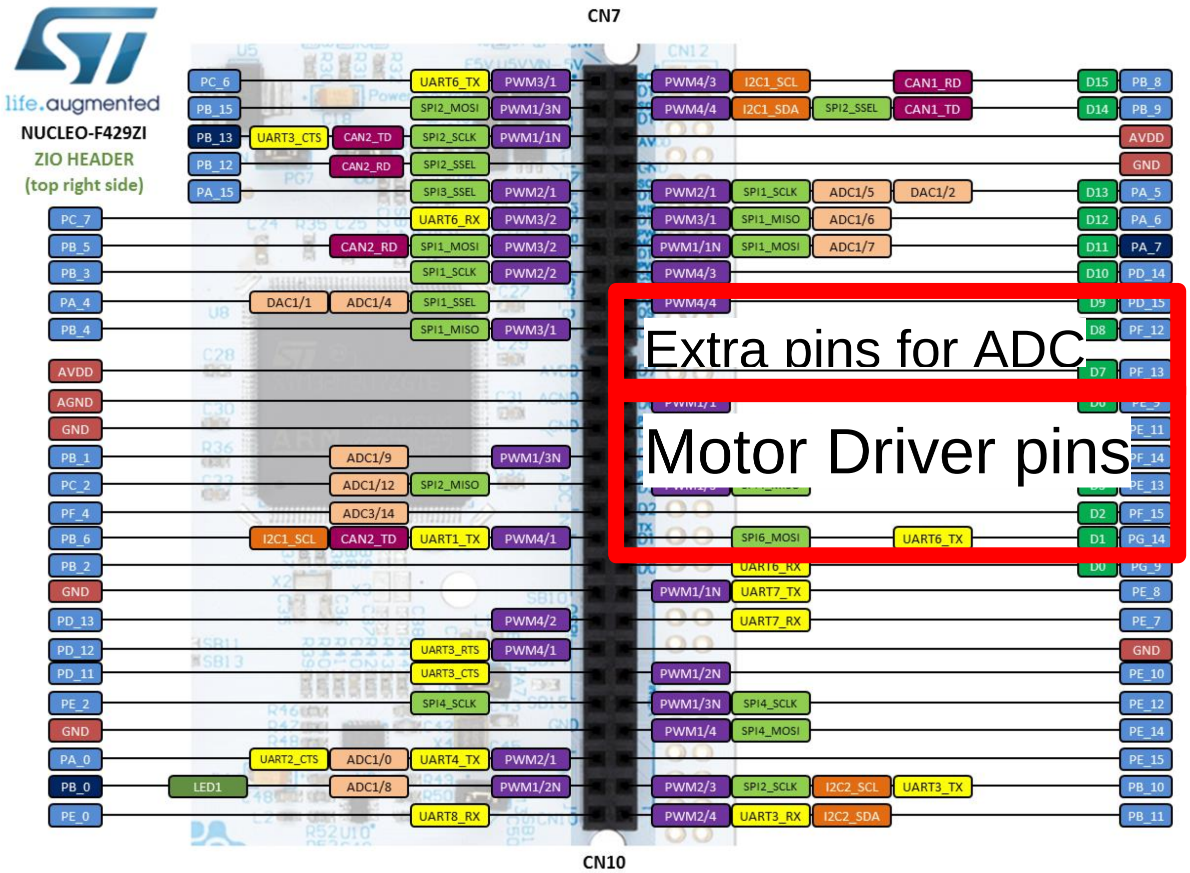


FIGURE 34 MCU PIN CONNECTIONS

Testing Laser sensor viability

The next step was to determine if the laser could measure the vibrations of an endmill. It is possible that either the surface finish or the geometry of the laser would scatter the light and the sensor would not be able to measure it. The sensor was tested in a configuration similar to a tap test. The sensor was placed inside a Haas VF-2 milling machine. Both an endmill and a solid shaft were tested in the milling machine. Since the shaft has no geometric features, it was used to verify if the signal was behaving differently. The tap used a normal hammer, and the force was not recorded. Figure 35 below shows the experimental setup.

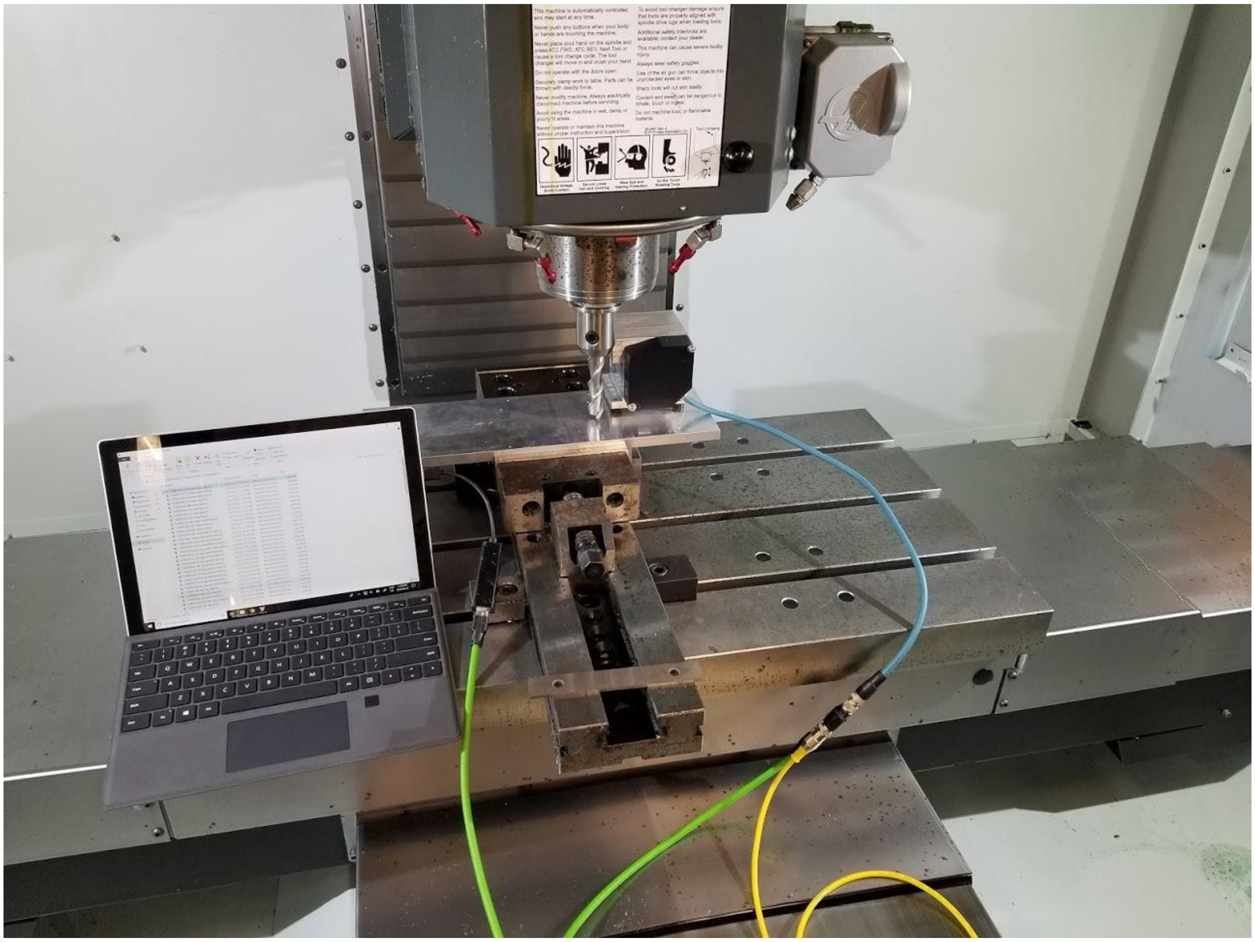


FIGURE 35 EXPERIMENTAL SETUP

The time data was transformed into frequency domain using a Fast Fourier transform. The time data was trimmed to create reduce noise in the frequency domain. The FRFs from the test are shown below. Figure 36 shows the FRF from taping just the shaft.

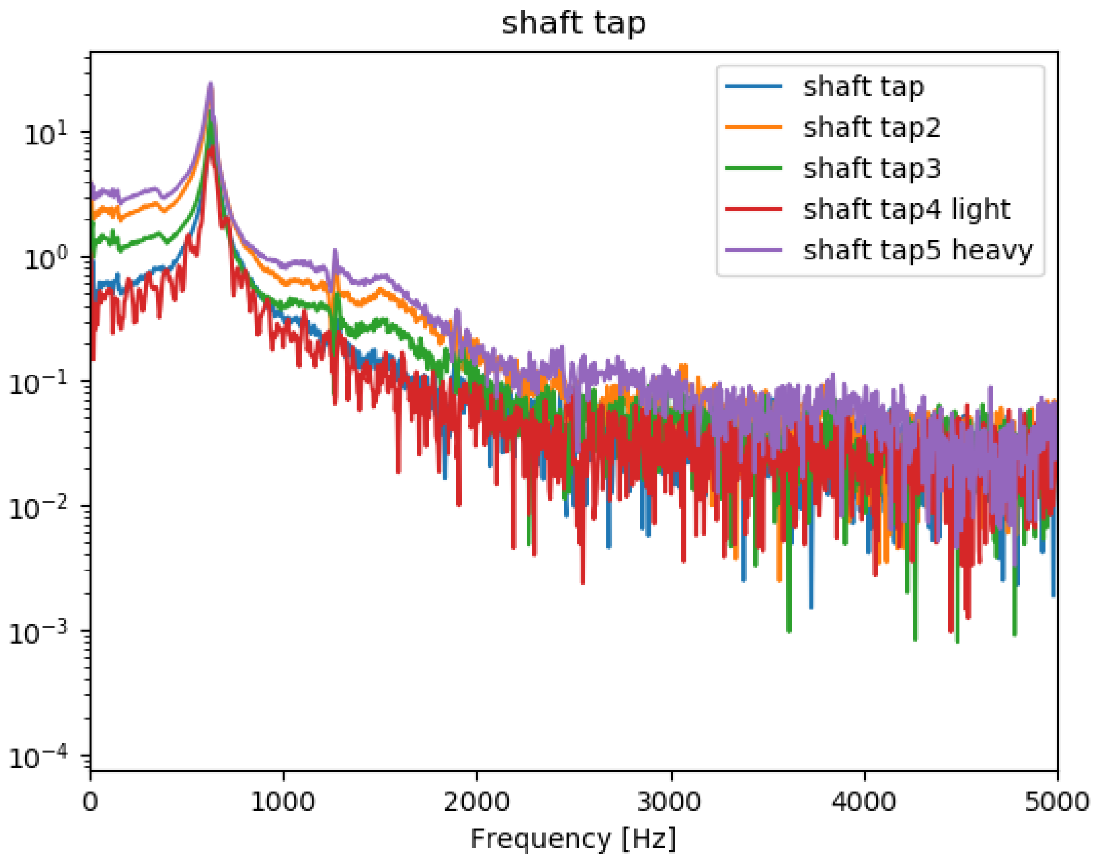


FIGURE 36 FRF FROM TAP TEST

The plots show multiple taps, and in each tap the FRF is fairly similar. There are slight differences in each tap, which accounts for some of the variability. However, there are distinct mode shapes that appear in all of the taps. This test was repeated using an endmill, and the results are in Figure 37 below.

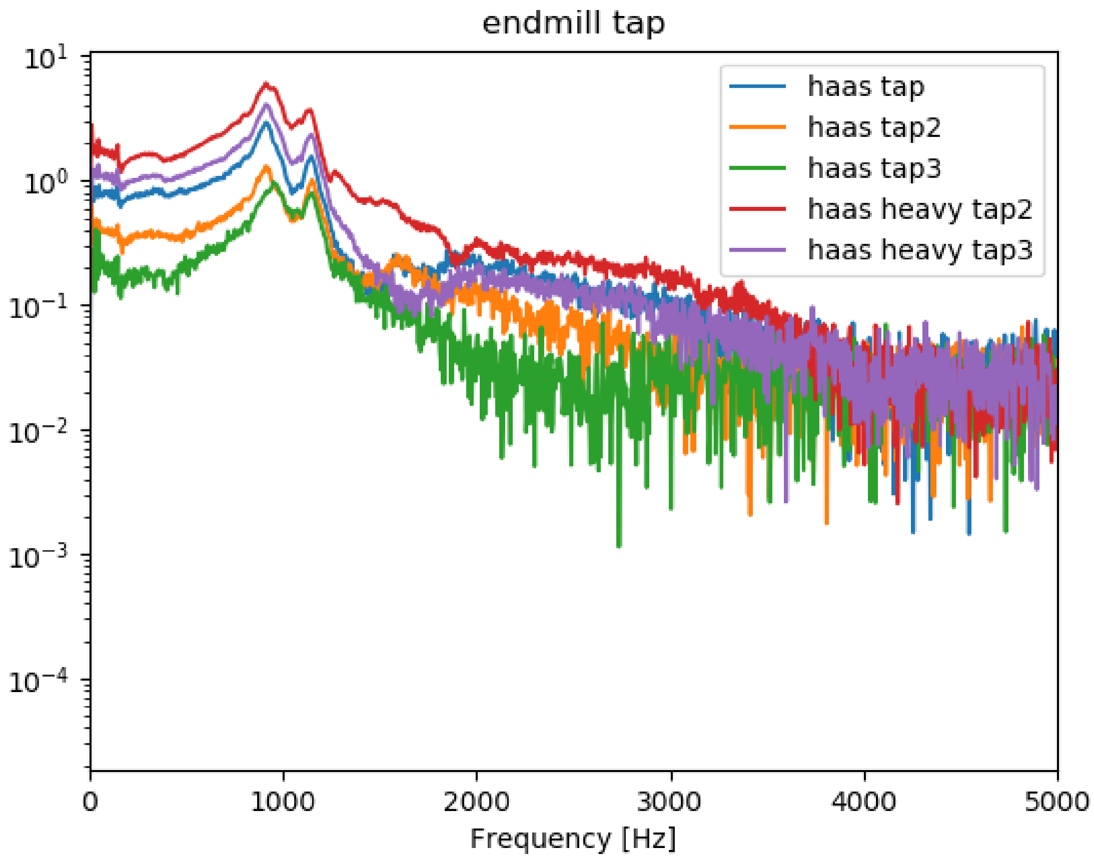


FIGURE 37 FRF FROM TAP TEST

In this FRF, the mode shapes are distinct from the previous test - this means that the FRF was the shaft vibrating, and not the laser vibrating. In both test cases, the bandwidth of the signal is at least 1000 Hz. The tap most likely did not excite high frequency content, thus the high frequencies contained noise. The sensor performed well with both the shaft and endmill.

Mount design

The automated tap tester should be a single unit that can be placed inside a machine. Both the voice coil and the laser will be mounted to a base plate. During an impact, the force goes both ways - into the endmill, and back into the hammer. Normally this is not a problem, since the hammer is in a human's hand. The human acts as a damper and does not transmit this force. However, in the case of the automated tap tester, the force can travel directly from the force sensor, though the voice coil and into the laser sensor. Since all of the components are metal and are fairly rigid, there is little attenuation. The components inside the laser senor can vibrate. Ideally the laser sensor would be isolated from the voice coil.

The initial design used a mass-spring-damper system to isolate the laser sensor. The spring damper system was purchased from newport. Figure 38 below shows the CAD model of the setup.

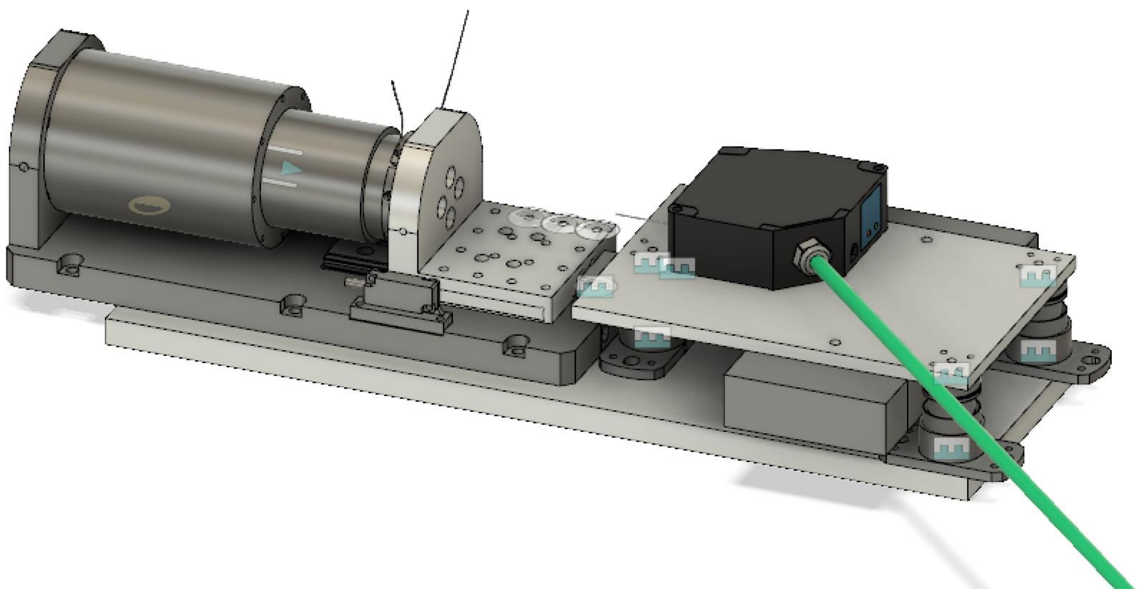


FIGURE 38 CAD MODEL OF AUTOMATED TAP TESTER

This prototype was built and tested. Unfortunately, it did not work as expected. The dampers provided a typical first order response, with a low frequency resonance. This isolated the laser from high frequency noise, but it also added a high low frequency component. This could have been accounted for in the FRF, however, the vibration isolated mount could move in all 6 axes. Because of this freedom, the response of the laser sensor became unpredictable. Rotations of the platform produced different results compared to translations. Furthermore, due to the large low frequency movement, it was also possible for the laser sensor to move outside its measuring range and lose signal. For these reasons, the vibration isolation mount was abandoned.

The alternative was to have a rigid mount for the sensor. The laser sensor would be mounted to a large piece of aluminum. The mass of the aluminum would absorb some of the impact. Unfortunately, testing showed that the internal components still vibrated from impacts using such a mount, but the vibrations were smaller than the expected endmill vibrations.

Tap testing using automated tap tester

The final step in verifying that the automated tap tester works would be to perform a tap test on a system with a known FRF. A tuning fork has known and easily excitable modes, and it was used for testing. A 512 Hz tuning fork was inserted into the tool holder in a TAIG milling machine. The voice coil and laser sensor were both mounted to a base plate, which was mounted to the table of the machine. Figure 39 below shows the experimental setup.

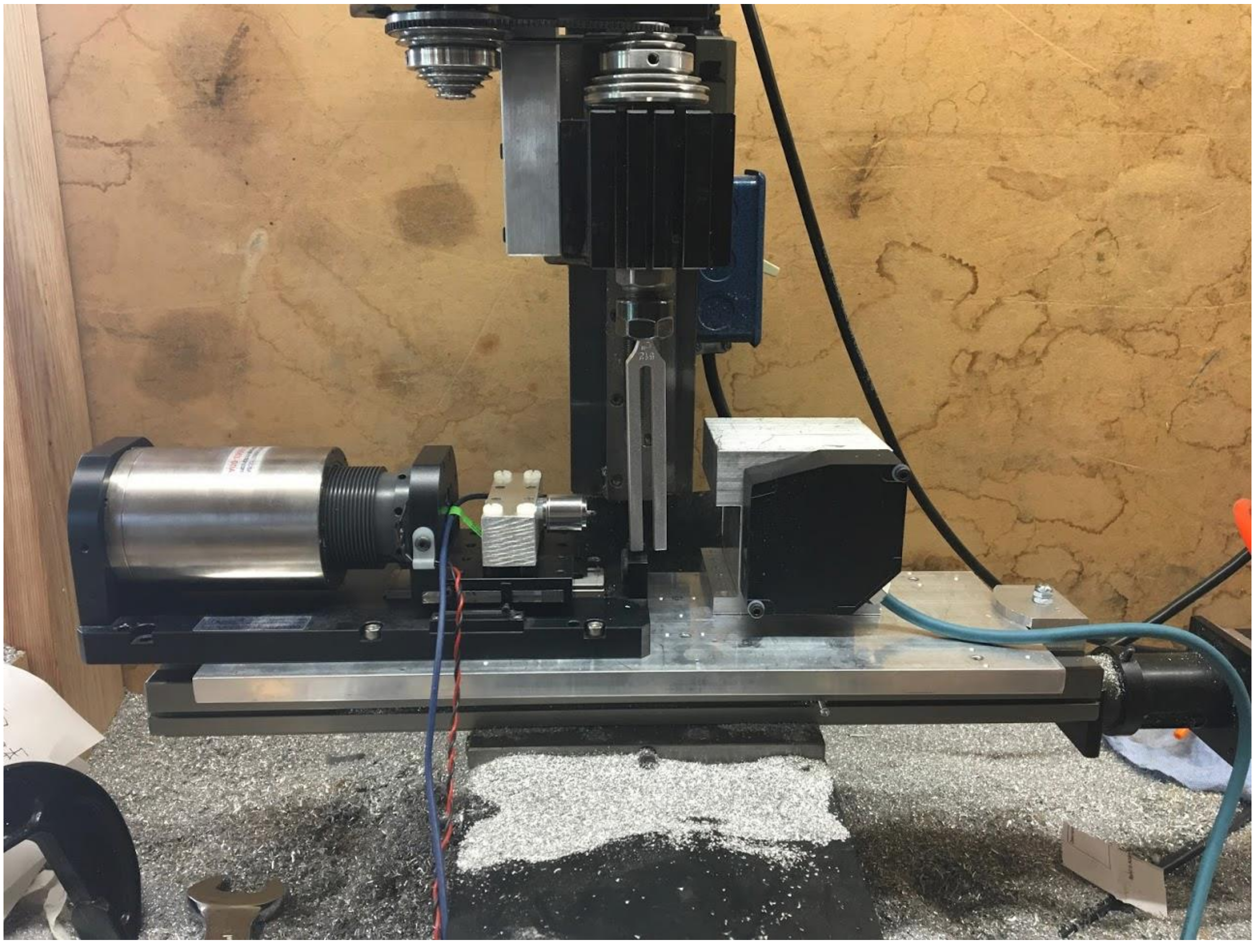


FIGURE 39 TUNING FORK TEST SETUP

The results from the tap testing this setup are summarized in the Figure 40 below.

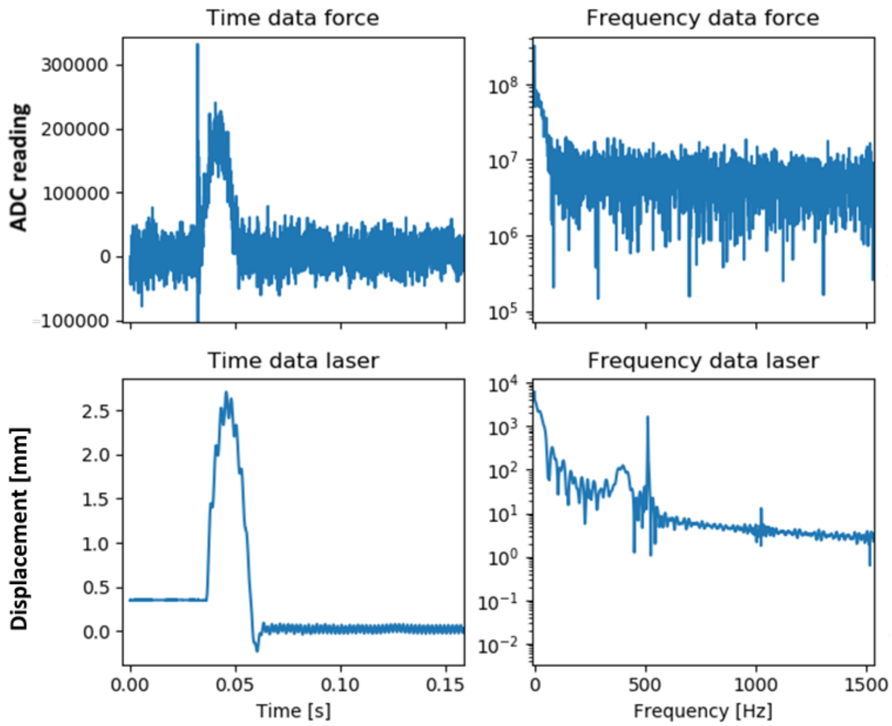


FIGURE 40 TUNING FORK TAP TEST RESULTS

Unfortunately, the results were sub-par. The voice coil was unable to impact the tuning fork and pull back in time. The time data of the fork shows an initial impact, and then a slow pull-back over approximately 20 milliseconds. This did excite the tuning fork, and the 512 Hz mode is visible, however, this would not work on endmills. The excitation bandwidth of the voice coil is too low.

Results

The automated tap tester did not satisfy the set-out requirements. However, it did prove that it is possible to build such a device. The alternative sensors (laser sensor and load cell) showed that they are capable of measuring a tap test. The limiting factor is making the voice coil excite the endmill with a high bandwidth tap.

The automated tap tester could measure the FRF directly since the laser sensor was at the same height as load cell. The actuation distance of the voice coil allows for various diameters of endmill to be used.

Furthermore, the laser sensor is a compact sensor that can measure vibrations at high frequencies, something that would be useful for measuring the FRF of small endmill.

Conclusion

To be able to predict when a machine will chatter, a dynamic model of said machine is needed. The current methods for making such a model are tedious and time consuming. An automated tap tester was proposed to automatically perform tap tests and determine the frequency response function of the machine and tool. The automated tap tester used the same principle as an impact hammer to measure the FRF - hitting the endmill and measuring the vibrations. Unlike the tap hammer, it was designed to be fully automated. The impact was delivered using a voice coil, and the vibrations were measured using a non-contact laser sensor. This allows the possibility to use the automatic tap hammer with a wide variety of endmills. Furthermore, measuring the vibrations with a non-contact displacement sensor allows for the tools to be swapped manually. This would potentially allow all of the tools in a machine to be tap tested without human intervention.

For the future

The failure of the voice coil is something that should be addressed in future projects. There are several possibilities as to why the voice coil did not perform well. The motor driver was rated for a lower voltage compared to the voice coil. The lower voltage means that the voice coil will accelerate slower. Furthermore, the driver might not have been designed to switch between such different velocities (full forward to full backwards). It is possible that the motor driver has some sort of slew rate that was not accounted for.

Conclusion

CNC machines are sophisticated and are a common manufacturing tool in industry. CNC machines have become incredibly sophisticated, yet they have no ability to make decisions. The machines can break tools and wreck parts and will continue to operate until manually stopped. This thesis addresses these issues.

Ideally, a machine would have enough sensors and intelligence to operate fully automatically. The machine would only need raw stock and the 3D part geometry, and would be able to figure out the toolpath and other parameters to make the part. Such sophistication is currently not possible. This thesis proposes several projects that take a step towards intelligent machines. An intelligent machine needs to be able to know what it is doing and make decisions based on that.

The first project involves using audio to determine the machine's state. A microphone records the audio while the machine is cutting, and a neural network determines if the machine is chattering or not. Unlike previous systems, the neural network is trained on raw data, and the data does not need to be processed.

The second project focuses on path planning. Modern CAM software generates tool paths, but it involves a lot of manual input. This project creates a toolpath planner that uses a model of the cutting process. The path it generates optimize for certain conditions while being physically possible. This allows the tool path to be generated without human intervention.

The last project does not focus on making CNC machines smarter, but on modelling them. Being able to predict how a machine will react to a certain input is the basis for artificial intelligence. Unfortunately current methods for determining the dynamic model of a CNC machine are tedious and require a lot of manual involvement. This project proposes an automated tap tester which can automatically determine the dynamic model of a machine. It will use a voice coil and a non-contact vibration sensor. This allows it to model all of the tools in a machine without human intervention.

All of the proposed projects are just one step in a long road to making fully autonomous CNC machines. The thesis projects were successful in limited ways. For integration into a smart CNC machine more steps and improvements will be required.

Bibliography

[1] Grand View Research, "Computer Numerical Control Machines Market, Industry Report, 2025," [Online]. Available: https://www.grandviewresearch.com/industry-analysis/computer-numerical-controls-cnc-market. [Accessed 10 09 2019].

[2] IBIS world, "Machine Shop Services in Canada - Market Research Report," [Online]. Available: https://www.ibisworld.com/canada/market-research-reports/machine-shop-services-industry/. [Accessed 10 9 2019].

[3] Y. Altintas, MANUFACTURING AUTOMATION, Vancouver: Cambridge University Press, 2012.

[4] V. S. O. V. d. A. E. F. P. D. Edouard Riviére, "Chatter detection techniques using microphone," 2006.

[5] Y. Y. X. C. Hongrui Cao, "Chatter detection in milling process based on synchrosqueezing transform of sound signals," Springer-Verlag, London, 2016.

[6] K. Z. X. C. X. Z. Hongrui Cao, "Early chatter detection in end milling based on multi-feature fusion and 3σ criterion," Springer-Verlag, London, 2017.

[7] D. Alvarez, "Usefulness of Artificial Neural Networks in the Diagnosis and Treatment of Sleep Apnea-Hypopnea Syndrome," InTech, 2017.

[8] P. Mathur, "A Simple Multilayer Perceptron with TensorFlow," 1 May 2016. [Online]. Available: https://medium.com/pankajmathur/a-simple-multilayer-perceptron-with-tensorflow-3effe7bf3466. [Accessed 10 9 2019].

[9] J. A. S. Hongcheng Wang, "A METRIC-BASED APPROACH TO 2D TOOL-PATH OPTIMIZATION FOR HIGH-SPEED MACHINING," ASME International Mechanical Engineering Congress & Exposition, New Orleans, 2002.

[10] K. C. H.S. Choy, "A corner-looping based tool path for pocket milling," Elsevier, 2001.

[11] S. W. F. J. a. J.-R. L. Qing-Hui Wang, "Adaptive trochoidal toolpath for complex pockets machining," International Journal of Production Research, 2016.

[12] T. B. Alexandru Dumitrache, "IMS10-image-based milling toolpaths with tool engagement control for complex geometry," Elsevier, 2012.

[13] V. S. I. Zaghbani, "Estimation of machine-tool dynamic parameters during machining operation through operational modal analysis," International Journal of Machine Tools & Manufacture, 2009.

[14] K. Dunwoody, "Automated Identification of Cutting Force Coefficients and Tool Dynamics on CNC Machines," THE UNIVERSITY OF BRITISH COLUMBIA, Vancouver, 2010.

[15] G. P. Gogue, "Voice-coil actuators," September 1989. [Online]. Available: http://www.consult-g2.com/papers/paper8/paper.html. [Accessed 22 October 2019].

[16] Manufacturing Automation Labratories, "Fundamentals of machining," 2013.

[17] E. B. Evren Burcu Kivanc, "Development of Analytical Endmill Deflection and Dynamics Models," ASME 2003 International Mechanical Engineering Congress and Exposition, 2002.