Real-Time Cutting Parameter Inference and Feed Rate Modulation by Machine Learning in CNC Milling of Wood

A Thesis Presented
by

Emilio Latorre

to

The Department of Information and Software Engineering

in partial fulfillment of the requirements
for the degree of

Master of Science

in

Cyber-Physical Systems

Northeastern University
Boston, Massachusetts

April 2026

I dedicate this thesis

To my mother and father, for their unwavering belief in and encouragement of me, not only during this thesis but also in every step of my education leading up to this.

To my grandmother, for her endless love and support, without which I would not have achieved this.

To both of my brothers, for listening and caring about my progress on this journey, and for putting up with me this far.

Un abrazo muy fuerte a todos.

Contents

List of Figures **iii**

List of Tables **iv**

List of Acronyms **v**

Acknowledgments **vi**

Abstract of the Thesis **vii**

1 Introduction **1**
- 1.1 CNC Processes 1
- 1.2 Cutting Parameters for CNC Processes 1
- 1.3 Wood as a Material for CNC Processes 4
- 1.4 Sensing and ML Applications to CNC Processes 5
- 1.5 Actuation of Cutting Parameters for CNC Processes 6
- 1.6 Objectives 6

2 Methods **7**
- 2.1 Sensor System Structure 7
- 2.2 Data Collection 10
- 2.3 Program Structure 11
- 2.4 Model Structure 12
- 2.5 Actuation System Structure 14

3 Results **16**
- 3.1 Sensing and Inference Testing 16
- 3.2 Actuation Testing 16

4 Conclusion **18**
- 4.1 Inference Performance 18
- 4.2 Actuation Performance 18
- 4.3 Future Improvements 19

Bibliography **20**

List of Figures

1.1 Diagram of the CNC mill, with axes labeled.[1] 2
1.2 Diagram illustrating depth of cut (ap) and stepover (ae)[2] 3
1.3 Diagram illustrating chip loading.[3] 4

2.1 Block diagram of the sensor system. 8
2.2 Schematic of the circuit of the sensor system. 8
2.3 The CNC mill with all sensors mounted. Note the placement of accelerometers (red), microphones (green), and current sensors (blue). 9
2.4 The CNC mill performing a test cut 10
2.5 A comparison of an optimal (bottom) and non-optimal (top) cutting result. 11
2.6 Average inference accuracy for linear regression by feature count 13
2.7 Block diagram of the actuation system structure. 14

List of Tables

2.1 Average inference times for different ML model types. 12

3.1 Results of confirmation testing using the final model. 16

List of Acronyms

CNC Computer Numerical Control.

ML Machine Learning.

MS MilliSecond.

CAM Computer-Aided Manufacturing.

G-code Programming language used with CNC machines.

RPM Revolutions Per Minute.

MEMS MicroElectroMechanical Systems.

Hz, kHz Hertz, Kilohertz.

FRO Feed Rate Override.

USB Universal Serial Bus.

SPI Serial Peripheral Interface.

I2S Inter-Integrated Circuit Sound.

ADC Analog to Digital Converter.

DFT Discrete Fourier Transform.

PC Personal Computer.

DC Direct Current.

API Application Programming Interface.

PD/PID Proportional-derivative, Proportional-Integral-Derivative.

KP/KD Proportional Constant, Derivative Constant (in context of PD/PID controller).

Acknowledgments

I would like to thank firstly my advisor, Dr. Rolando Herrero. His encouragement, advice both technical and rhetorical, and insight into the thesis process made this project possible. His knowledge not only of the technical aspects of such varied subjects as carpentry and control theory, but also of the process itself of producing and defending the thesis proved invaluable, and without him I would not have been able to complete this project. I would also like to acknowledge professors Dr. Abhishek Murthy and Andrew King for their opinions and suggestions at various points in the development of this thesis. I am grateful to my father, Emilio Latorre Sr., for his assistance on some of the finer points of woodwork and for fostering a lifelong love of the art in me at an early age. I would like to thank professors Dr. Sergey Aityan, Dr. Haitham Tayyar, and Dr. Peter O'Reilly for serving on my defense committee.

Abstract of the Thesis

Real-Time Cutting Parameter Inference and Feed Rate Modulation by Machine Learning in CNC Milling of Wood

by
Emilio Latorre
Master of Science in Cyber-Physical Systems
Northeastern University, April 2026
Dr. Rolando Herrero, Advisor

Computer numerical control milling is the most precise and efficient method of manufacturing many types of wood parts. Wood is a highly variable organic material, which presents challenges to the operation of milling machines. Although cutting parameters for metals are well-established, setting parameters in wood requires a higher degree of operator observation and skill. We present a system that implements real-time inference of cutting parameter optimality on a computer numerical control mill using a sensor constellation and machine learning inference. The machine learning model is computationally lightweight and can perform inference at a high rate, giving constant feedback as to whether cutting parameters are set optimally. In live testing with real cutting operations, our model correctly infers over 94% of samples while maintaining sampling and inference times under 2 milliseconds. This data is then fed back into a proportional-derivative controller which modulates the machine feed rate, implementing closed-loop optimization of cutting parameters without operator intervention.

Chapter 1

Introduction

1.1 CNC Processes

CNC tools have been in wide use in industrial and production manufacturing settings for several decades [4]. CNC machines perform manufacturing operations specified by a program. The program is written in G-code, either directly by an operator or using CAM software, describing the desired operations. The program specifies movements such as linear moves and arcs, as well as machine state updates such as turning spindles, coolant pumps, and other tools on and off. The controller of the CNC machine executes these instructions to produce a part. CNC is applicable to a wide variety of machining tools and processes, most commonly mills and lathes. The particular tool used in this system is a 3-axis mill. The cutting tool is a spindle head which holds a router bit. The workpiece is clamped to a flat table under the spindle. The chassis comprises 3 axes, two of which move the spindle and one of which moves the table, together allowing 3D motion of the bit relative to the workpiece. An example unit of the specific model of machine used in this system is shown in Figure 1.

1.2 Cutting Parameters for CNC Processes

When cutting with rotary tools such as router bits, there are several cutting parameters that must be set. Optimization of these parameters is a crucial factor in maintaining precision and surface quality of the produced part.[5] Depth of cut refers to the length of the section of the cutter that engages the workpiece in each cutting pass. Stepover refers to the fraction of the diameter of the

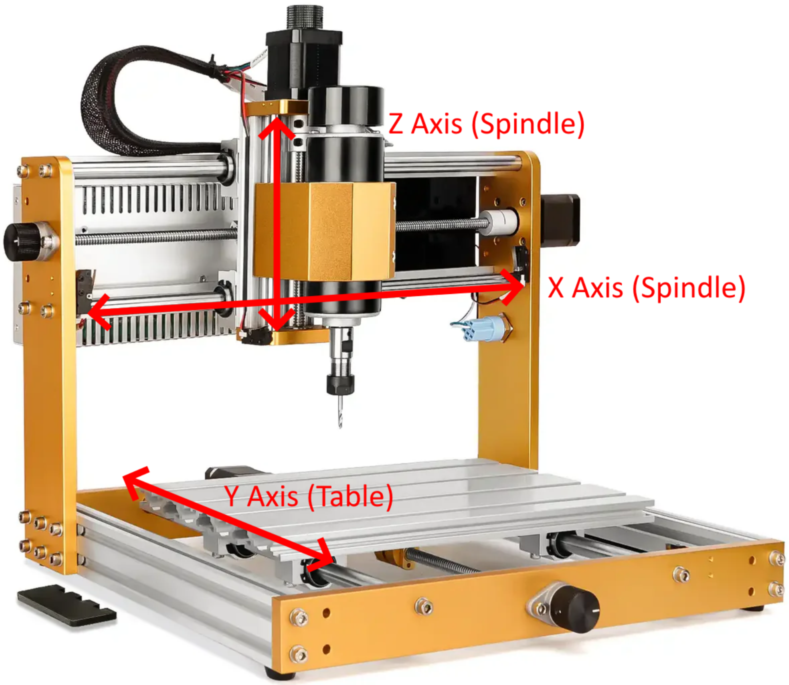

Figure 1.1: Diagram of the CNC mill, with axes labeled.[1]

cutting tool that is engaged in each cutting pass. A diagram illustrating depth of cut and stepover is shown in Figure 2.

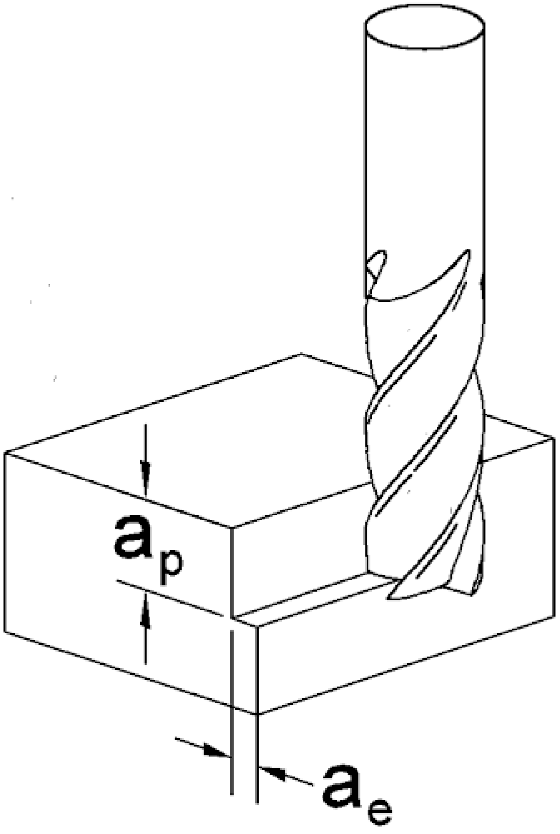

Figure 1.2: Diagram illustrating depth of cut (ap) and stepover (ae)[2]

These must be set in the programming phase as changing them involves changing the path of the cut. Because of this, they are not suitable for live monitoring and control. Feed rate is the speed at which the axes move and is in units of a linear distance per unit of time. Rotational speed is the rate at which the cutter spins and is in units of revolutions per unit of time. By combining the number of flutes on the cutting tool with the feed rate and speed, it is possible to calculate chip loading, which is the amount of material removed each time a cutter flute passes the material.[3] An example is shown in Figure 3, with the blue area corresponding to chip loading.

Although these parameters are set in the initial program, modifying them does not change the path of the tool and thus they can be updated while the machine is running. This makes them suitable for real-time feedback to the operator. Setting correct chip loading is necessary to optimally

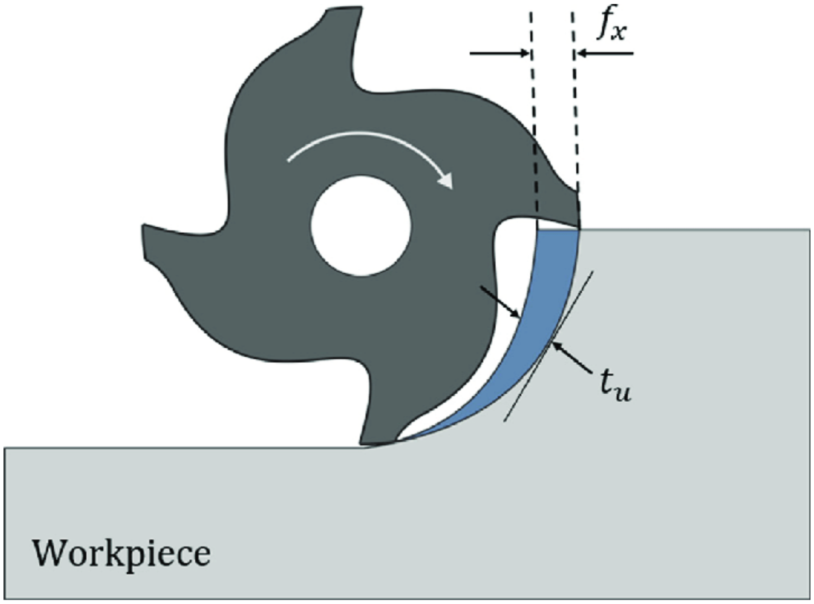

Figure 1.3: Diagram illustrating chip loading.[3]

minimize vibration, tool and workpiece heating, and cycle time. Incorrectly setting feed rate too high increases cutting forces on the bit, which causes excess vibration, tool chatter, deviations from the programmed tool path, and reduced tool life.[6] This increased stress also increases wear on the machine. Setting feed rate too low also reduces tool life and can cause issues in surface finish due to increased heating from friction as the tool rubs against the workpiece without cutting.

1.3 Wood as a Material for CNC Processes

Recently, the falling price of hardware has put CNC tooling within reach of smaller manufacturers and hobbyists. Many of these users will be working primarily with wood. The cutting parameters of machine tools in metals are extensively studied and well-specified. It is common for manufacturers to directly specify optimal cutting parameters for a given cutter and type of metal.[7] Wood, however, is an extremely variable category of materials. It differs in properties not only from species to species, but also between different pieces of stock of the same species. It is affected by environmental factors such as storage humidity and temperature, and is highly anisotropic, with cutting difficulty varying relative to grain orientation.[8] Inclusions and discontinuities such as knots, voids, and cracks are frequent. These factors combine to make the selection of cutting parameters much more variable than with metal.[5] These factors must be accounted for by the operator of the machine, and typically require significant expertise on their part to set and monitor correctly. Cakiroglu et al. and Sofuoglu determined that for machining wood with small cutters, optimal spindle speed is at or above 10,000 RPM.[5, 9] As this is near the RPM limit of the machine used here, this leaves feed rate as the parameter by which chip loading can be modulated. Li et al. and Kim et al. describe systems in which the feed rate of a CNC tool is modulated actively during cutting operations.[5, 10]

1.4 Sensing and ML Applications to CNC Processes

The use of sensors to monitor CNC machines is well-studied. Li et al., Kim et al., and Wszołek et al. describe the application of MEMS accelerometers to monitoring the vibration of a CNC machine.[6, 10, 11] Vibration is one of the primary mechanisms by which incorrect cutting parameters affect cut quality, and thus accelerometers are an important sensor for monitoring a CNC machine. Additional sensors such as dynamometers,[6] current sensors,[10] and microphones[12] are also utilized in some systems as additional data sources. Li et al. use a dynamometer coaxially mounted to the spindle to measure the force applied to the cutting tool between the workpiece, providing an indication of how hard the cutting tool is being driven.[6] As chip loading is raised, increasing the amount of material being cut with each pass, the force required to cut this material also increases. Input from the dynamometer was used in time domain, as the fluctuations are dynamic with changes in cutting path[6]. Kim et al. use current sensors, which measure the current drawn by the spindle motor.[10] This provides another metric of the force being applied to the workpiece, as the current drawn by an electric motor increases as the torque it applies increases with resistance applied to it. Similar to the dynamometer, this input was also considered in time domain. In all of these systems, inputs from accelerometers and microphones are considered in frequency domain rather than time domain. Relevant vibration and audio signals will usually be periodic rather than single spikes, so it is most useful to consider these signals in the frequency domain. This periodicity corresponds to the frequency of rotating components in the machine, such as the spindle motor and axis lead screw motors. Sampling frequencies on the order of 1 kHz[11] to 10 kHz[6] are demonstrated. These high sample rates are necessary for the sampling frequency to exceed the Nyquist sampling rate of important vibration signals, such as the first and second harmonics of the rotational rate of the spindle. Various methods are used to interpret sensor data, ranging from relatively simple curve-based algorithms[10] to ML models such as neural networks.[13]

1.5 Actuation of Cutting Parameters for CNC Processes

Various systems have been described which include modulation of cutting parameters in response to sensor inputs. The system used by Li et al. modulated feed rate using a 2-step process, in which the feed rate is first set in the program sent to the machine using a physics simulation.[6] During cutting operations, inputs from the sensor constellation of an accelerometer and dynamometer are fed into a fuzzy logic control process which modulates the feed rate within a limited band.[6] Kim et al. use a constellation of an accelerometer and current sensor on the spindle motor.[10] Their system uses a response curve to translate sensor inputs into modulation of the programmed feed rate. FRO is a system by which the feed rates programmed in G-code are modified in the machine controller, with speeds increased or decreased. This feature is very commonly supported by CNC machines, especially mills. Li et al. and Kim et al. utilize this method to modulate feed rates at runtime.[6, 10]

1.6 Objectives

In light of the aforementioned considerations about CNC processes in general, and the CNC milling of wood in particular, it is clear that setting cutting parameters requires a significant degree of operator skill, and is affected by the inconsistencies of wood as a material. We propose a system to firstly analyze whether cutting parameters have been set optimally, and secondly to set them optimally at runtime, without operator intervention. Together, these eliminate the need for operator skill in setting correct feed rates in G-code programming and monitoring them during cutting operations.

Chapter 2

Methods

2.1 Sensor System Structure

The system consists of two hardware subsystems, the sensor system and the control system. The sensor system collects data on machine state from a sensor constellation and feeds it to the ML model where it is used to make inferences about the optimality of current cutting parameters. The structure of the sensor system is outlined in Figure 2.1, and the circuit diagram of the system is shown in Figure 2.2.

The sensing system consists of a sensor constellation connected to the PC by USB. The sensor system is controlled by an RP2350A microcontroller. The RP2350A is connected to three ADXL345 three-axis accelerometers over an SPI bus and two INMP441 microphones over an I2S bus. Two WCS1800 current sensors are connected directly by analog output to the RP2350A ADC inputs, as their carrier modules only provide a digital threshold output driven by a voltage comparator rather than a true digitized output. The RP2350A streams data from the sensors to the PC over an emulated serial port on a USB connection. The sensors are mounted to the CNC mill. The first accelerometer is mounted on the spindle carriage, the second on the top bar of the Y-axis, and the third on the workpiece table. One microphone is mounted on the Z-axis, and the other on the top bar of the Y-axis. One current sensor is mounted to measure the current consumed by the spindle, and the other to measure the current consumed by the power supply to the whole CNC mill. The placement of these sensors is shown in Figure 2.3.

The firmware on the microcontroller handles polling of the sensors and communication to the PC. This link is unidirectional, and the microcontroller does not require any input from the PC. The accelerometers and current sensors are polled at 1 kHz. The microphones are polled at 40 kHz.

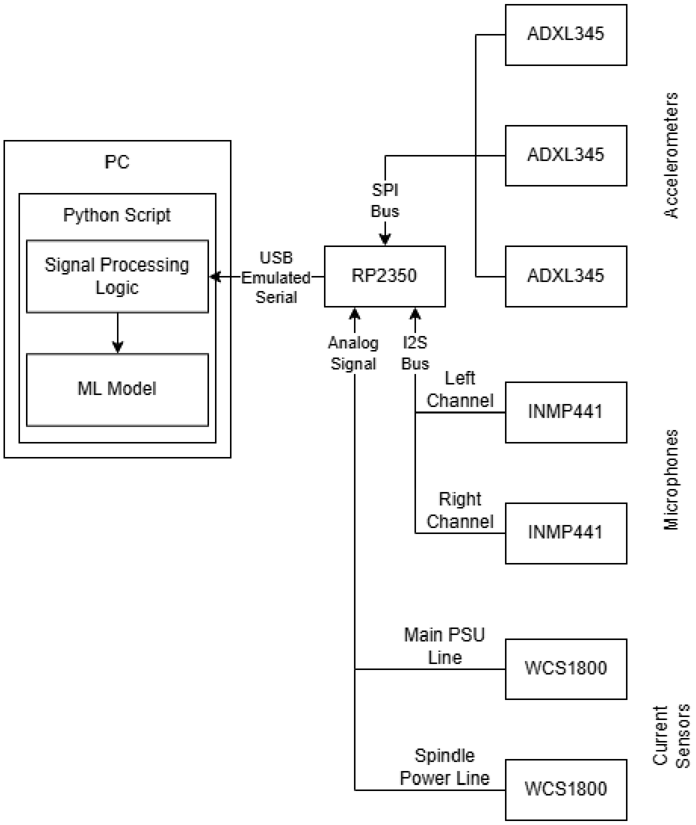

Figure 2.1: Block diagram of the sensor system.

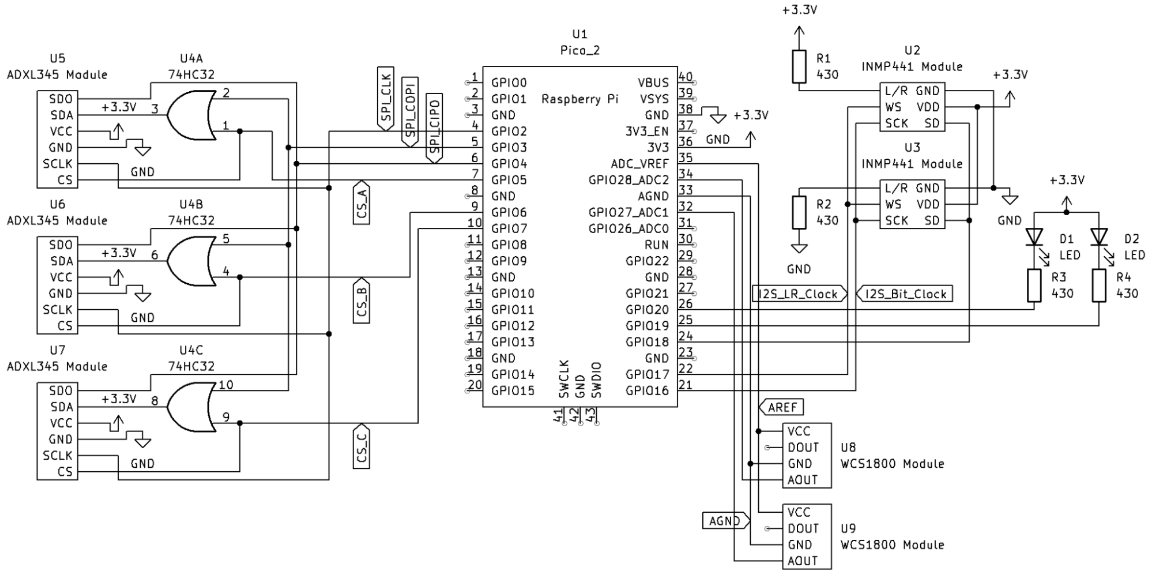

Figure 2.2: Schematic of the circuit of the sensor system.

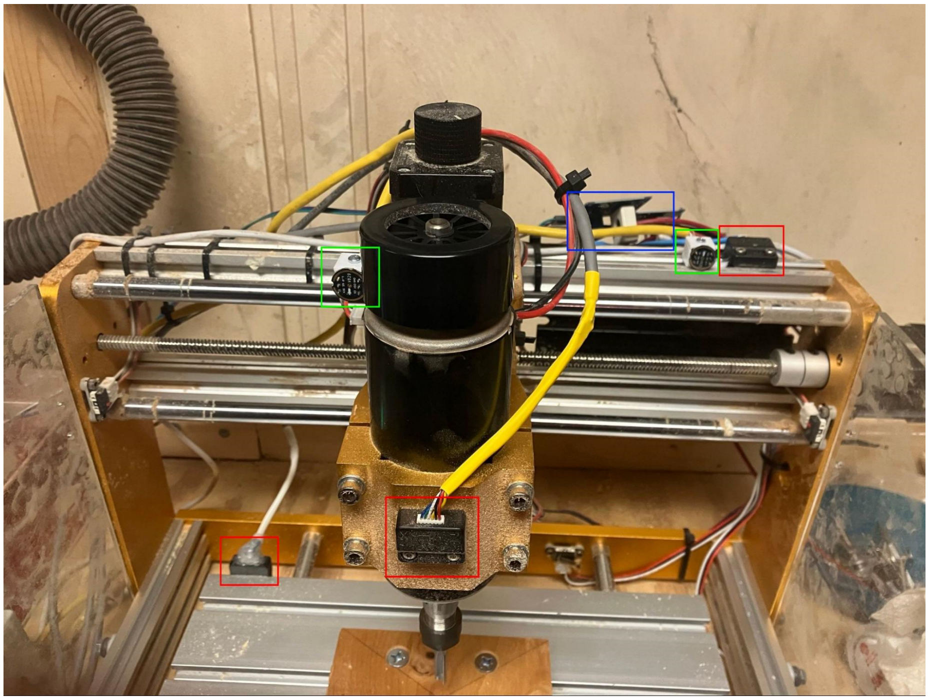

Figure 2.3: The CNC mill with all sensors mounted. Note the placement of accelerometers (red), microphones (green), and current sensors (blue).

The spindle does not have a tachometer but has a design RPM of between 11,000 RPM and 12,000 RPM at maximum speed, equivalent to 183.3 Hz to 200 Hz. In data gathering, there is consistently a high vibration energy at 190 Hz, confirming this approximate speed. The lower sampling rate of 1 kHz is sufficient to capture vibrations of the spindle, as this exceeds the Nyquist sampling rate of the second harmonic of the spindle rotation. The higher sampling rate of 40 kHz exceeds the Nyquist sampling rate of the approximate limit of audible sound. At a rate of once per millisecond, the 40 most recent audio samples are packed into a struct with the current acceleration and current readings. This struct is then sent over the serial connection to the PC, where it is received by the Python script on a virtual serial port. The use of packetization and burst transmission ensures that the virtual serial port is not overloaded and reduces time lost to transmission overhead, while still ensuring that the transfer rate is sufficiently fast to allow for real-time inference.

2.2 Data Collection

The test workpieces used were all pine boards, 3/4 of an inch thick, and mounted to the table using 4 screws with tee nuts. All cutting operations were conducted with a 1/4 inch 2 flute straight router bit. Figure 2.4 shows a workpiece fixtured in the mill in the experimental configuration.

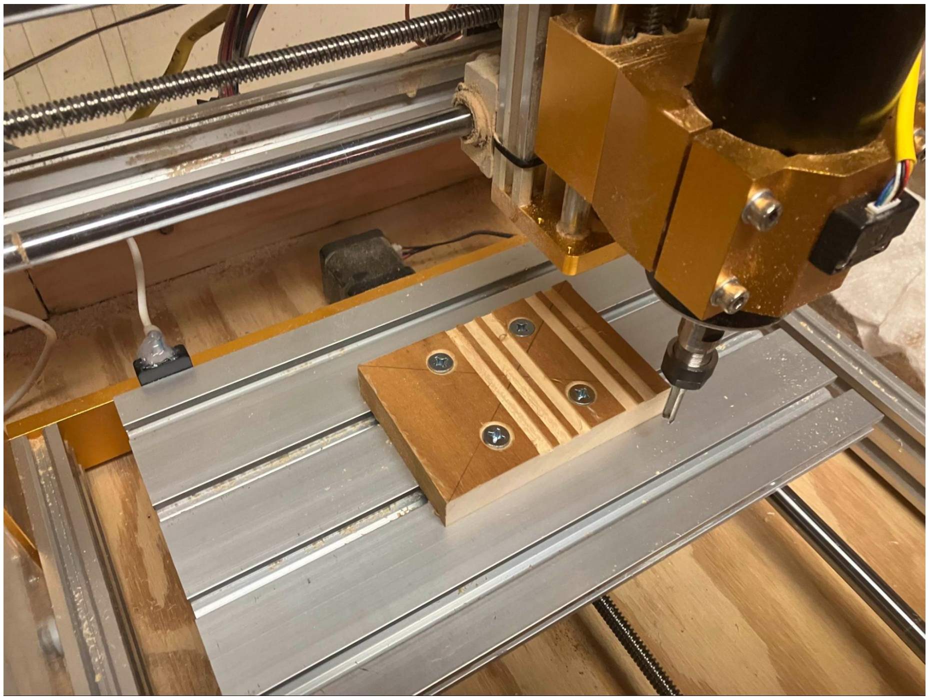

Figure 2.4: The CNC mill performing a test cut

Approximately 30 cutting operations were recorded for the training data set. Approximately half of the cuts were oriented in line with the grain of the wood, and half were oriented at 90 degrees to the grain, testing both ripping and crosscutting operations. A range of cut depths from 1mm to 5mm were tested. Samples of approximately three seconds of data were recorded from each of these cutting operations. This data was recorded and combined into a training data set for use in the ML model. These data were used for supervised training of ML models. The categorization was performed manually by observing the quality of the final cut surface and visually classifying the quality as optimal or non-optimal. Examples of optimal and non-optimal quality cuts are shown in Figure 2.5.

Note that although the manual categorization performed here relies on operator skill, the specific categorization between optimal or non-optimal is irrelevant to the accuracy or lack thereof of

Figure 2.5: A comparison of an optimal (bottom) and non-optimal (top) cutting result.

the control system, as long as it is consistent within the training data set. An arbitrary categorization could be selected, and the model could be trained accurately to that categorization. A final training data set consisting of 23,704 observations was collected and recorded for training ML models using this configuration. Later confirmatory testing of the models would utilize the same configuration.

2.3 Program Structure

The program on the PC is written in Python and runs on the PC at the time of use. It captures all incoming data from the RP2350A's emulated serial port. Incoming data packets are parsed by the Python script as they come in at a rate of 1ms between packets. Once the packet is parsed, the raw data of the single acceleration and current samples and 40 audio samples are inserted into ring buffers holding data from the last 1 second, corresponding to 1,000 samples each for acceleration and current data and 40,000 samples each for audio data. As described in section 1.4, the current data is treated directly as a time series, and thus is not processed further before it is passed to the ML model. The acceleration and audio data is converted to frequency domain before it is passed to the ML model. Once per inference call to the ML model, a DFT is performed on the buffered samples to generate frequency-domain data. The window for this DFT is set at 100 samples for acceleration data and 8192 samples for the audio data, representing a time window of 100ms for the acceleration DFT and 205ms for the audio DFT. These sampling rates and window sizes combine to give the computed DFT a spectrum of 5Hz-20000Hz divided into 4096 bins for audio data, and 10Hz-500Hz divided into 50 bins for the accelerometer data, after dropping the DC component bin for both. Overall this results in 1 feature × 2 channels of current data, 50 features × 9 channels of acceleration data, and 4096 features × 2 channels of audio data, resulting in an overall feature vector of 8644 features.

The processed feature vector is then fed into an ML model which produces a boolean inference of the current cutting conditions as optimal or non-optimal. The model is loaded separately from the main program and presents an inference API to the script, allowing for the testing and use of different models while maintaining the same program structure. When gathering data for training, this data is logged to a file on the PC. When the program is being used for confirmatory testing or feed rate control, this data is fed into the model to infer conditions at the maximum possible rate. When used for testing, the program logs the stream of inferences to a file. When used for feed rate control, the inferences are passed to the actuation system.

2.4 Model Structure

The total vector of data features used for both training and inference is 8644 features. This consists of 2 channels of current data with one feature each, 9 channels of acceleration frequency-domain data with 50 features each, and 2 channels of frequency-domain audio data with 4096 features each. The 2 current features are treated as time-domain signals. For other sensors, the number of features are derived from the number of frequency bins in the DFT applied to the incoming signals. In order to maintain a consistent input API for the predictive model, this feature vector is held constant regardless of the model's internal structure. Different models used at different points in testing use subsets of this data internally, and do not utilize all features in the incoming samples in the generation of inferences. The models all ingest the same full vector of 8644 features and internally discard unused features and/or perform compression before inference.

It was determined that an ML model would be used to infer whether the incoming data represented an optimal or non-optimal cutting operation. Several ML model types were evaluated for performance, both in time and accuracy. For all training runs, each model was trained on 80% of the data and tested for inference accuracy against a randomly selected 20% holdout test. This process was then repeated to calculate average accuracy and inference time.

Model Type	**Average Inference Time**
Linear Regression	0.332 ms
Polynomial Regression (Degree 2)	602 ms
Neural Network (Hidden Layers 1x64 Nodes)	1.62 ms

Table 2.1: Average inference times for different ML model types.

The linear model vastly outperformed both other models in inference speed. The polynomial regression model runs in O(N2), and thus is not suitable for real-time inference. The neural network runs in O(N) time, but took longer than the linear model on average by a factor of over 4. The linear regression also runs in O(N) time and was fastest by far, and was thus selected as the model type to be used for testing the effect of feature compression.

Within each model type, different levels of feature compression were tested. Although the inference API used by all models was kept constant with the full feature vector as an input, internally varying levels of compression reduce the number of features used in the actual inference. This compression is achieved by simple averaging of the DFT bins to produce the desired number of features. Having determined linear regression to be the most efficient model in time, the effect of feature compression on accuracy was tested more extensively with this model. After testing, it became apparent that the microphone features were not necessary to produce accurate inferences. Thus, the final linear model discards all microphone features. The single current feature per channel was retained, as it does not require a DFT and adds virtually no computational cost, and thus there is little benefit from its removal. The linear regression was retrained using first 2 features per acceleration channel, then increasing up to 30 features per channel. The results of this testing are shown in Figure 2.6.

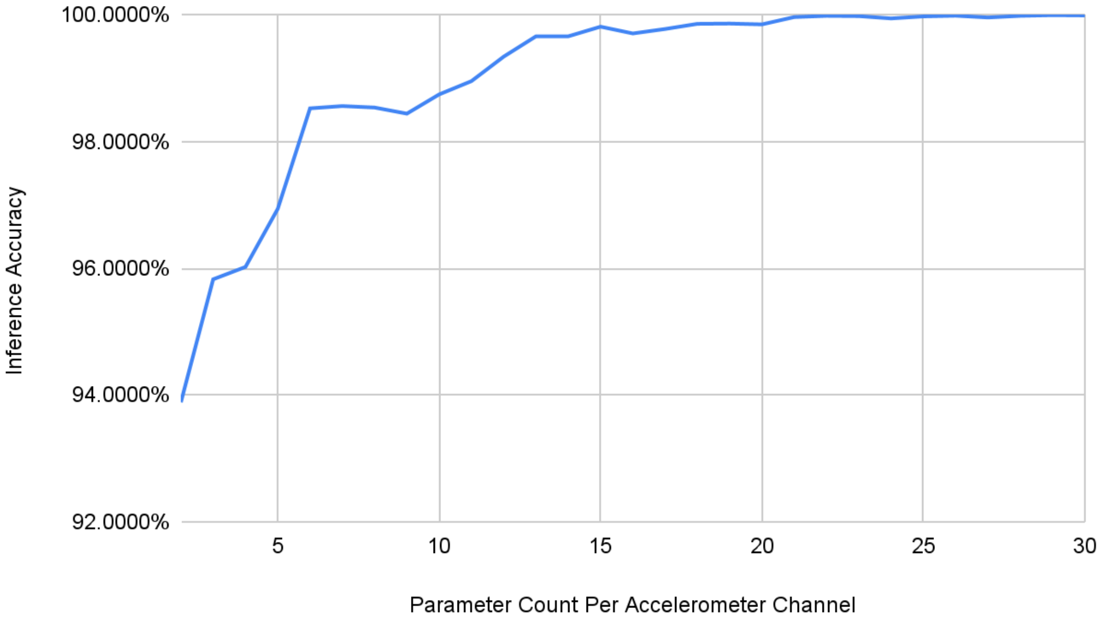

Figure 2.6: Average inference accuracy for linear regression by feature count

Using the linear regression, only 12 features per channel are required to exceed 99% inference accuracy, with accuracy leveling off and fluctuating around 99.6% as feature count is increased further. This keeps the model lightweight, and greatly reduces the memory and computational resources theoretically needed to run the inference model within real-time constraints.

2.5 Actuation System Structure

The second subsystem is the actuation system. The actuation system uses inferences from the ML model as an input to actively modulate the feed rate of the CNC mill. The structure of the actuation system is outlined in Figure 2.7.

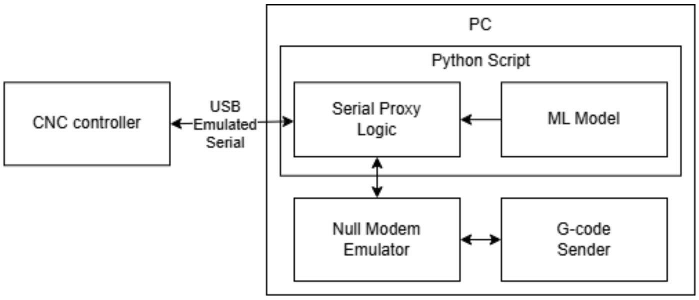

Figure 2.7: Block diagram of the actuation system structure.

In standard operation, the CNC controller directly receives G-codes from a G-code sender application on a PC over an emulated serial port connection. Using a software null modem emulator running on the PC, commands from the G-code sender are instead routed to a serial port opened by the Python script. This is the same Python script used to run the signal processing logic and ML model in the sensing system. The commands are then read and forwarded by the serial proxy logic in the Python script to another serial port, which connects to the CNC controller's USB emulated serial port. Most commands are forwarded directly from the null modem to the CNC controller without modification.

The CNC controller exposes a set of FRO commands by which the operator can manually alter feed rate at runtime during standard operation. The FRO is applied as a percentage by which to multiply the programmed feed rate; for example, at a FRO of 50%, a programmed move with a feed rate of 1000mm/min would be executed at 500mm/min. This controller exposes commands for FRO up 1% and FRO down 1%. FRO commands sent by the G-code sender are discarded by the serial proxy logic and not forwarded.

Inputs to the serial proxy logic from the ML model are combined with a control algorithm to instead send FRO commands based on inferred cutting conditions. It was selected to use a PD controller to modulate feed rate. As inferences are boolean and the PD controller requires a variable input, inferences from the last 1 second are buffered, and the fraction of samples inferred optimal is used as the input to the PD controller. In addition to the PD control, it was found during testing that rapid oscillation of feed rate produces undesirable cutting artifacts. Additionally, as high feed rate is more dangerous than low feed rate, it is necessary for the model to be able to modulate feed rate down more quickly than up. To correct for these considerations, a slew rate limit was applied to the feed rate modulation, with a significantly lower rate limit for modulating up than modulating down.

Chapter 3

Results

3.1 Sensing and Inference Testing

Final testing was performed using the linear regression, trained with no microphone features, 15 features per acceleration channel, and 1 feature per current channel. Tests were performed using the same setup as when collecting training data. Feed rate modulation by the actuation system was not enabled during these tests. Collected data was recorded alongside the inferred optimality and then compared by visual inspection to the actual cut. Results of these tests are shown in Table 2.

Metric	**Value**
Total Samples	2474
Average Samples per Test	165
Average Inference Accuracy	94.61%
Average Inference Time	1.787 ms

Table 3.1: Results of confirmation testing using the final model.

Over all tests, the average inference accuracy was 94.61%, with an average inference time of 1.787 ms.

3.2 Actuation Testing

After confirmatory testing of inference performance, the actuation system was also tested. Manual tuning of the PD controller and slew rate parameters was performed. A KP of 0.8 and KD of 0.03, along with a slew rate of 15% per second (modulating up) and 220% per second (modulating down) were found to produce acceptable cutting quality and reduced oscillation in feed rate. With these parameters, the system produced optimal cutting results even when programmed with a feed rate that in initial testing produced non-optimal results.

Chapter 4

Conclusion

4.1 Inference Performance

In practical application, the sensing and inference component of the system performed well. The accuracy was reasonably high, although lower on average than in backtesting during training. We propose 2 likely sources for this discrepancy. Firstly, because the tests are manually labelled by inspection of the cut result, it is possible that the samples within a test were not entirely optimal or non-optimal, but instead contained some samples of both. Additionally, it is likely inclusions in the wood such as knots may have caused such a change in cutting conditions during a portion of the cut. This level of inference accuracy is still sufficiently high as to be useful for driving the actuation system. The inference speed performance, while similarly lower than in backtesting, was still good. The practical inference rate was approximately 500 Hz, which given the speeds of the system involved is still sufficiently fast to drive an appropriate reaction time in the actuation system.

4.2 Actuation Performance

In practical application, the actuation system performed acceptably. Through the application of automated feed rate modulation, the requirement for the user to set the correct feed rate in the G-code programming stage is removed. The user can simply select a very high feed rate and allow the automated control system to reduce it to an appropriate level.

4.3 Future Improvements

There are several avenues by which the performance of the overall system might be improved. In the ML model, it would be useful to improve the accuracy of inferences and the accuracy of the confirmatory testing. In both the training data and confirmatory testing stages, it would be useful to have some way of matching specific data samples to specific locations on the cut path. One option for this would be to use a synchronized video recording to track where along the cut path the head is at the time a particular sample was recorded. In the model itself, it may be that although a highly compressed feature vector was sufficient for high accuracy in backtesting, a higher number is needed for practical application to novel data. It would be useful to repeat the confirmatory testing with a model with a less compressed feature vector. Our findings show that only a relatively small number of features per channel are needed to accurately infer cutting condition optimality. Additionally, the microphone features proved to be unnecessary, which greatly reduces the size of the feature vector without any compression. In future implementations, rather than calculating such a large feature vector and compressing it within the model, it would be possible to directly calculate a simpler DFT of 32 bins per channel over the 500 Hz spectrum of the acceleration and eliminate the 4096-bin per channel microphone DFTs. This would vastly reduce the computational and memory resources needed to run the inference engine within real-time constraints, and may allow for implementation of inference on the RP2350 or a more powerful constrained device rather than on the PC. All tests were conducted with pine boards from the same lot. Although this provides a significant variety in terms of grain orientation and knot inclusion, this means that species, moisture content, and board quality were kept constant. It would be useful to conduct further testing with different species of wood and different moisture contents or grades of board.

Bibliography

[1] LUNYEE Industries, "Lunyee 3018 pro max cnc router 500w all-metal," https://www.lunyeecnc.com/products/lunyee-3018-pro-max-cnc-router-laser-2-in-1-machine, 2025, accessed: 2026.

[2] Toolnotes, "Cutting parameters for milling," http://toolnotes.com/home/machining/mills-101/cutting-parameters-for-milling/, 2026, accessed: 2026.

[3] L. Pang, A. Hosseini, H. Mohamed, and I. Deiab, "Application of a new thick zone model to the cutting mechanics during end-milling," *International Journal of Mechanical Sciences*, vol. 96–97, pp. 91–100, 2015.

[4] A. Ullah, T.-C. Chan, and S.-L. Chang, "Current trends in vibration control and computational optimization for cnc machine tools: A comprehensive review," *The International Journal of Advanced Manufacturing Technology*, vol. 139, pp. 5409–5444, 2025.

[5] E. O. Çakıroğlu, A. Demir, and İsmail Aydın, "Determination of the optimum feed rate and spindle speed depending on the surface roughness of some wood species processed with cnc machine," *Journal of Anatolian Environmental and Animal Sciences*, vol. 4, no. 4, pp. 598–601, 2019.

[6] Y. Li, Z. Zhao, K. Wang, N. Qian, Y. Fu, and S. Cao, "Active-passive hybrid feed rate control systems in cnc machining: Mitigating force fluctuations and enhancing tool life," *Journal of Manufacturing Systems*, vol. 77, pp. 184–195, 2024.

[7] Kennametal, Inc., "Gomill™ gp • square end • 2 flutes • weldon® shank • inch," https://www.kennametal.com/us/en/products/p.gomill-gp-square-end-2-flutes-weldon-shank-inch.6086461.html, 2025, accessed: 2026.

[8] A. Cakmak, A. Malkocoglu, and S. Ozsahin, "Optimization of wood machining parameters using artificial neural network in cnc router," *Materials Science and Technology*, vol. 39, no. 14, pp. 1728–1744, 2023.

[9] S. D. Sofuoglu, "Determination of optimal machining parameters of massive wooden edge glued panels which is made of scots pine (pinus sylvestris l.) using taguchi design method," *European Journal of Wood and Wood Products*, vol. 75, pp. 33–42, 2017.

[10] S.-G. Kim, E.-Y. Heo, H.-G. Lee, D.-W. Kim, N.-H. Yoo, and T.-H. Kim, "Advanced adaptive feed control for cnc machining," *Robotics and Computer-Integrated Manufacturing*, vol. 85, p. 102621, 2024.

[11] G. Wszołek, P. Czop, J. Słoniewski, and H. Dogrusoz, "Vibration monitoring of cnc machinery using mems sensors," *Journal of Vibroengineering*, vol. 22, no. 3, pp. 735–750, 2020.

[12] M. Iqbal and A. K. Madan, "Cnc machine-bearing fault detection based on convolutional neural network using vibration and acoustic signal," *Journal of Vibration Engineering & Technologies*, vol. 10, pp. 1613–1621, 2022.

[13] W.-L. Chu, M.-J. Xie, Q.-W. Chang, and H.-T. Yau, "Research on the recognition of machining conditions based on sound and vibration signals of a cnc milling machine," *IEEE Sensors Journal*, vol. 22, no. 7, pp. 6364–6377, 2022.