Characterizing Distinct CNC Machining Signals to Enhance Process Control

Katherine Niemeyer
University of Colorado - Boulder

ABSTRACT

Sensors are widely used in manufacturing control to track tool performance and quickly react to processing abnormalities. Because of the noisy nature of production data, it can be difficult to reliably apply machine learning techniques to characterize in-situ signals. This paper aims to develop a signal processing workflow that quickly characterizes Computer Numerical Control (CNC) machining signals for a given operation, as well as distinguish OK from NOK signatures.

Utilizing an operation-specific processing strategy before transforming each signal led to a Support Vector Classifier that achieved 98.6% accuracy on training data, and 98.0% accuracy on validation data. Furthermore, the same classifier showed that for 32 known bad samples, only 3 (9.4%) of those would be mistakenly classified as a 'good' sample from the same operation.

1 INTRODUCTION

Signal analysis is a large area of research in the manufacturing space, as connected data from processing tools is constantly transmitted to production databases. The challenge is to interpret these signals in real-time and leverage this information for process control. One area of production signal analysis is Computer Numerical Control (CNC) machining, where 3-dimensional accelerometers are attached to the machine spindle to transmit acceleration data as milling operations are performed. These acceleration signals are transferred to a database; however, quality auditors must still manually classify milled parts as good (OK) or bad (NOK).

CNC machining sensor data presents an opportunity to deconstruct each operations' unique signature, and use this model to automatically classify a part as OK or NOK. However, current research is limited to laboratory experiments, which do not capture the breadth of processing and environmental variables within production data.

1.1 Project Proposal

This project proposes a signal processing methodology and resulting classification model that is able to detect unique milling operations, and within each operation, classify OK vs NOK samples.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND AND RELATED WORK

Accelerometers are used on CNC machines to track the movement of the tool while undergoing a drilling operation. The 3-dimensional accelerometer traces can then be decomposed and classified into unique operations, where each drilling operation has a unique signature that represents an 'OK' run. However, after each batch an expert on the shop floor controls the resulting workpiece in a gauging station and annotates the process health [1]. Because quality auditors are still responsible for checking each milled part, this slows down processing times and also represents an opportunity for human error.

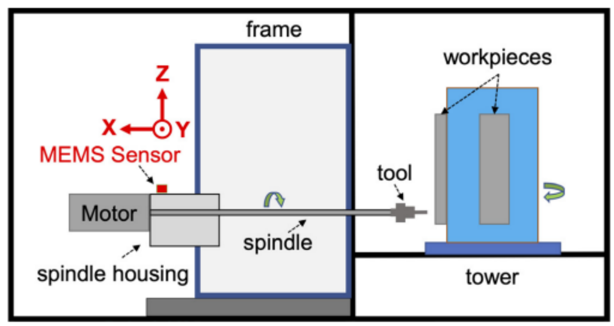


Figure 1: CNC Accelerometer Setup [1]

This work is primarily an extension of the research reported by Tnani et al. in *Smart Data Collection System for Brownfield CNC Milling Machines: A New Benchmark Dataset for Data-Driven Machine Monitoring* [1]. Their work focused on providing the software architecture required to stream CNC accelerometer data, and presented an overview of signal processing techniques. The authors' main goal was to publish the Bosch CNC production dataset for others to expand upon, in the hopes of creating robust machine learning models that can handle noisy production data. As previously stated, most research on machining data has been performed in a laboratory setting, with the two main contributors being NASA [2] and the University of Michigan SMART Lab [3].

This project will use the dataset published by Tnani et al., which consists of 2 years worth of CNC milling data from 3 different tools. The tools each process 15 unique operations, and contain labels for OK and NOK samples.

3 METHODOLOGY/DESIGN

3.1 Methodology Overview

Signal Processing

Each sample in the Bosch CNC Milling dataset consists of x, y, and z acceleration traces. This work will focus on the recorded x-axis traces to reduce complexity within the dataset. Additionally, the source dataset contains 15 operations, however 12 of those are step drilling tasks. The scope of the analysis was limited to 6 different step drilling operations (OP00, OP01, OP03, OP04, OP05, OP06), as well as the three unique operations (OP02: Drill, OP09: Straight Flute, and OP13: T-Slot Cutter).

Tool operation	Description	speed [Hz]	feed [$mm\ s^{-1}$]	duration [s]
OP00	Step Drill	250	≈ 100	≈ 132
OP01	Step Drill	250	≈ 100	≈ 29
OP02	Drill	200	≈ 50	≈ 42
OP03	Step Drill	250	≈ 330	≈ 77
OP04	Step Drill	250	≈ 100	≈ 64
OP05	Step Drill	200	≈ 50	≈ 18
OP06	Step Drill	250	≈ 50	≈ 91
OP07	Step Drill	200	≈ 50	≈ 24
OP08	Step Drill	250	≈ 50	≈ 37
OP09	Straight Flute	250	≈ 50	≈ 102
OP10	Step Drill	250	≈ 50	≈ 45
OP11	Step Drill	250	≈ 50	≈ 59
OP12	Step Drill	250	≈ 50	≈ 46
OP13	T-Slot Cutter	75	≈ 25	≈ 32
OP14	Step Drill	250	≈ 100	≈ 34

Figure 2: Selected CNC Operations [1]

From each selected operation, a single sample is analyzed to better understand the characteristics of the signal. A low pass filter is defined to maximize signal clarity, then the sample is trimmed to a single repeat unit. This process is repeated for each operation, then applied to all respective samples in the dataset. The Fast Fourier Transform (FFT) is then computed, and the resulting output is stored in an array, ready to be analyzed.

Dimensionality Reduction and Clustering

After initial preprocessing of the data is complete, the dimensionality must be reduced prior to training a clustering algorithm. To do so, the Fourier transformed signals will then undergo Principal Components Analysis (PCA) to reduce each signal into its primary features.

The decomposed matrix is then passed into a clustering routine to separate and classify each of the 9 operations. The clustering algorithm must be able to distinguish OK samples that belong to different operations, and the expectation is that a successful algorithm will show that NOK signals do not match their operation labels.

3.2 Details on Techniques

Data Loading

The CNC_Machining GitHub repository [4] was cloned, allowing access to the 3D signals saved as .h5 files.

Signal Cleaning

First, a low pass filter (LPF) was selected and applied after iterating through multiple cutoff frequencies, with the goal of minimizing noise while maintaining key peaks within the signal. Once the signal was de-noised, a combination of scipy.signal's *find_peaks* and *peak_prominences* functions were used to isolate the periodic component within a trace. This was done by iteratively tuning a peak prominence threshold (PROM TH) that highlighted the strongest peaks within the sample. The resulting peak indices were then used to define a range that fully described one repeat unit within the signal (PEAK RANGE). The sequence was then trimmed by selecting the sample array within these indices. The final step in the cleaning routine was to scale all the amplitudes to [-1,1] using sklearn's *MinMaxScaler*.

Operation	LPF	PROM TH	PEAK RANGE	STEP TYPE
OP00	1.5	400	3-7	Step Drill
OP01	10	1000	12-24	Step Drill
OP02	10	1000	0-8	Drill
OP03	3	150	2-23	Step Drill
OP04	2	500	1-3	Step Drill
OP05	20	1500	3-6	Step Drill
OP06	7	1000	6-11	Step Drill
OP09	2	400	1-5	Straight Flute
OP13	20	1000	1-10	T-Slot Cutter

Table 1: Signal Cleaning Hyperparameters: Low Pass Filter, Peak Prominence Threshold, and Peak Range for Trimming

After the cleaning hyperparameters were established based upon a sample of each operation, this method was globalized to all samples in the dataset. The final tuned cleaning parameters are included for each selected operation in Table 1.

However, due to the varying processing times between operations, not all trimmed signals were the same length. To combat this issue, each sample was padded to the maximum array length, then FFT was applied using the *scipy.fft* module. The real component of each FFT output was saved to an array.

By transforming the signals using FFT, each operation is now able to be compared frequency-wise. Padding each signal to the same length, then analyzing the absolute value in the frequency domain allows for direct comparison of signals with different magnitudes and time scales, which is crucial for the robustness of the solution.

Dimensionality Reduction

After transforming each good signal in the dataset, the resulting matrix had shape (993x76170). This is far too many dimensions to efficiently cluster on, so Principal Components Analysis (PCA) was used to determine the 5 primary features within the trimmed and transformed dataset.

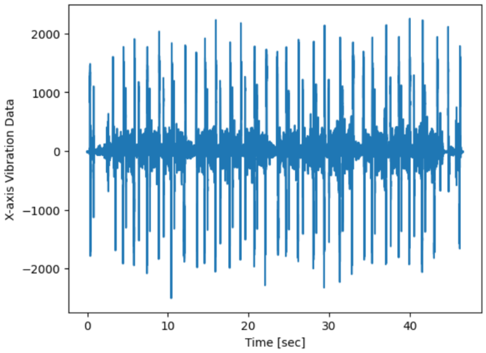
Clustering

After dimensionality reduction, two clustering techniques were compared: KMeans and a Support Vector Classifier. KMeans clustering is easy to implement given the known 9 distinct operations. However the data must be linearly separable for the algorithm to work. Using a support vector classifier allows for some leniency in this condition, because multiple hyperplanes can be used to separate the data according to known classes.

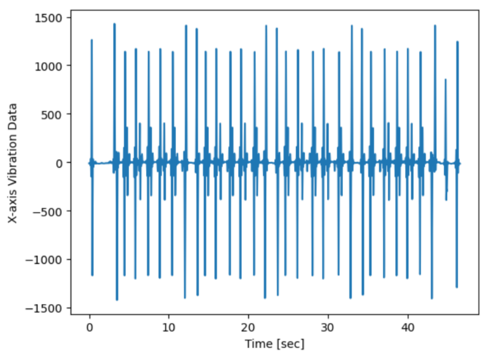
4 RESULTS

As previously stated, the effectiveness of the signal processing workflow and resulting classification model is primarily based upon its accuracy and resulting confusion matrix. First and foremost, the model must be able to distinguish good signals between distinct operations.

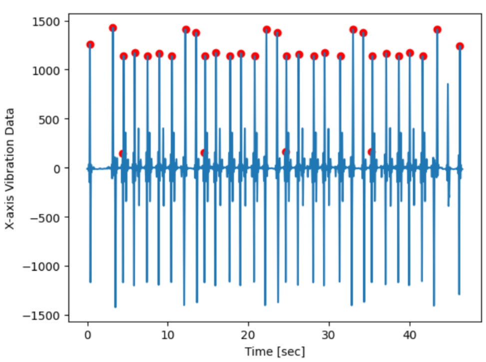
Secondly, the goal of the model is to reduce manual classification of good vs. bad samples by a quality auditor. Therefore, the model should show that within an operation, known bad samples do not match their good counterparts. This is assessed based upon



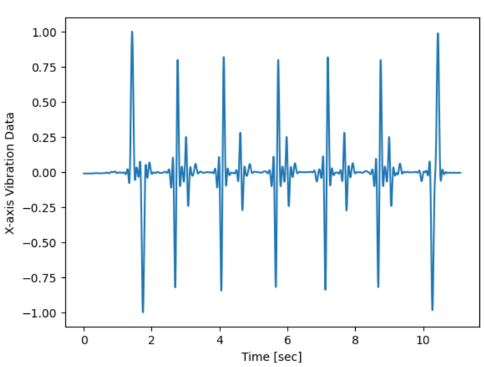
(a) Raw Signal



(b) Signal After 20 Hz Low Pass Filter



(c) Filtered Signal with Peaks Highlighted



(d) Trimmed and Scaled Signal

Figure 3: Signal Cleaning Steps from (a) Raw Signal, (b) Filtered, (c) Peak Identification, and (d) Trimmed to One Repeat Unit

a confusion matrix of bad samples across all operations, and a good model will classify these NOK signals on the off-diagonal.

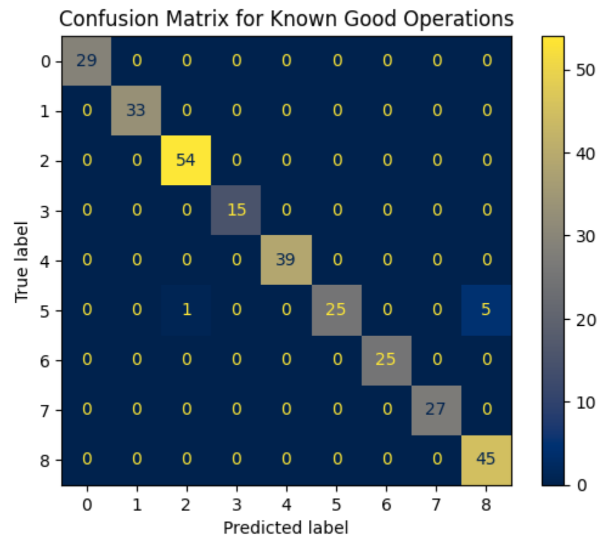


Figure 4: Confusion Matrix on Good Test Signals

The top performing clustering model was a Support Vector Classifier (SVC) trained on 30% of the good signals across the 9 selected operations. This model, which uses balanced class weights and the default rbf kernel, resulted in a classification accuracy of **98.6%**. The witheld validation data was classified with 98% accuracy. The confusion matrix of the test results is included in Figure 4. The only true class with incorrect predictions was OP05, which had the shortest periodicity at 4 seconds.

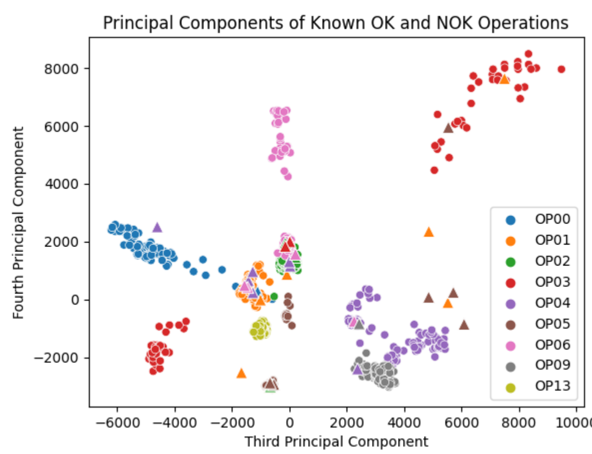


Figure 5: Comparison of Principal Components for Known OK and NOK Signals by Operation

Interestingly, KMeans clustering performed poorly on the dataset, with a maximum accuracy of 65%. This is most likely due to linear inseparability of the clusters along the principal axes. This is illustrated in Figure 5, which had the best visual separation between clusters across the third and fourth prinicpal components. Even still, the different operations show significant overlap, leading to the poor performance of KMeans.

Figure 5 also shows how the NOK signals, colored according to their true labels, compare with OK signals. In many cases, the NOK signals' principal components do not align with the good signals from the same operation, and instead visually overlap with different clusters. This effect is also seen when NOK signals are passed to the SVC: the resulting confusion matrix only bins the NOK signals according to their true labels 3/32 times [Figure 6]. This shows that NOK signals do not align with the expected behavior of the accelerometer at that operation.

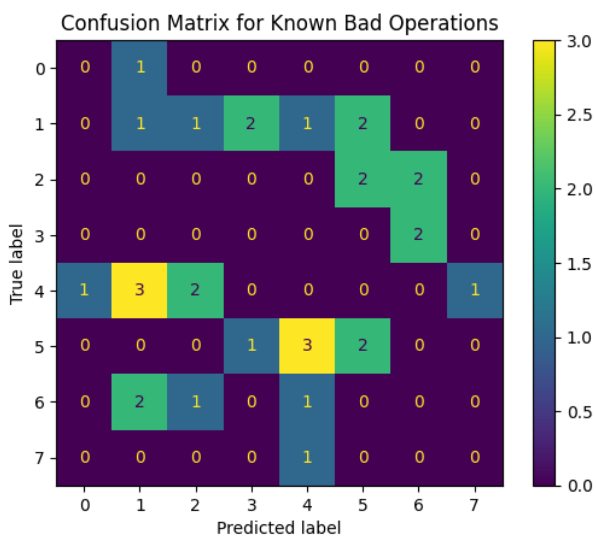


Figure 6: Confusion Matrix of Known Bad Signals

5 DISCUSSION

5.1 Challenges and Further Work

One of the main challenges encountered during the analysis was determining how to format the data to be ready for a machine learning model. The signals have ramp up and ramp down regions, and are not the same length, so developing a subsetting strategy that could be globalized took some time. To overcome these challenges, each of the operations have hand-tuned trimming procedures, and afterwards, they are all padded to the same length and transformed into the frequency domain. Also, since each operation needed a custom cleaning procedure, only half of the original 12 step drilling operations are included in the analysis.

In future work, it would be interesting to see if readily available windowing functions perform as well or better than the operation-specific tuning strategy used here. Additionally, one could compare the false positive rates obtained by training specific models for each operation versus a global classifier that is used across all milling steps.

5.2 Timeline

This project is largely dependent on the development and tuning of a robust signal processing algorithm. Therefore, that stage of the analysis required the most time to complete. The goal timeline for this project was the proposal completed by Week 3, with the signal processing algorithm tuned by the end of Week 5. In actuality, the signal processing algorithm was tuned by the end of week 6. Tuning the clustering model took minimal time in comparison and was completed in the beginning of week 7, allowing for completion of the final project and presentation by week 8.

6 CONCLUSION

Sensors are widely used in manufacturing to capture process data and map batch-to-batch variation. However, production data streams are notably noisy. Operational and environmental variables can mask true signals, limiting the effectiveness of machine learning's use for process control.

By implementing a signal processing methodology that decomposes CNC accelerometer data into primary frequencies, manufacturing noise is reduced, enabling the use of classification models. This paper presents a signal processing workflow that leverages filtering and trimming to obtain a characteristic repeat unit for each milling operation.

This pre-processing methodology allows for the use of Support Vector Classifiers to differentiate unique operation signals with **98% accuracy**, as well as quickly distinguish between OK and NOK samples without relying on quality auditors alone.

7 REFERENCES

[1] Mohamed-Ali Tnani, Michael Feil, Klaus Diepold, *Smart Data Collection System for Brownfield CNC Milling Machines: A New Benchmark Dataset for Data-Driven Machine Monitoring*, Procedia CIRP, Volume 107, 2022, Pages 131-136, ISSN 2212-8271, https://doi.org/10.1016/j.procir.2022.04.022.

[2] A. Agogino and K. Goebel. Best lab, uc berkeley. milling data set, nasa ames prognostics data repository. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/, 2007.

[3] System level Manufacturing and Automation Research Testbed (SMART) at the University of Michigan. Cnc milling dataset. https://www. kaggle.com/shasun/tool-wear-detection-in-cnc-mill, 2018.

[4] Bosch Research, CNC Machining, (2022), GitHub repository. https://github.com/boschresearch/CNC_Machining.git

[5] Harden, S. W. (2020, September 23). Signal filtering in Python. Signal Filtering in Python. https://swharden.com/blog/2020-09-23-signal-filtering-in-python/