

A Data Processing Pipeline for Prediction of Milling Machine Tool Condition from Raw Sensor Data

M. Ferguson¹, R. Bhinge², J. Park³, Y. T. Lee⁴, and K. H. Law¹

Abstract

With recent advances in sensor and computing technology, it is now possible to use real-time machine learning techniques to monitor the state of manufacturing machines. However, making accurate predictions from raw sensor data is still a difficult challenge. In this work, a data processing pipeline is developed to predict the condition of a milling machine tool using raw sensor data. Acceleration and audio time series sensor data is aggregated into blocks that correspond to the individual cutting operations of the Computer Numerical Control (CNC) milling machine. Each block of data is preprocessed using well-known and computationally efficient signal processing techniques. A novel kernel function is proposed to approximate the covariance between preprocessed blocks of time series data. Several Gaussian process regression models are trained to predict tool condition, each with a different covariance kernel function. The model with the novel covariance function outperforms the models that use more common covariance functions. The trained models are expressed using the Predictive Model Markup Language (PMML), where possible, to demonstrate how the predictive model component of the pipeline can be represented in a standardized form. The tool condition model is shown to be accurate, especially when predicting the condition of lightly worn tools.

¹ Stanford University, Civil and Environmental Engineering, Stanford, CA, USA

² Infinite Uptime, Inc. Berkeley, CA, USA

³ Korea Advanced Institute of Science and Technology, Industrial and Systems Engineering, Daejeon, South Korea

⁴ National Institute of Standards and Technology, Systems Integration Division, Gaithersburg, MD, USA

1 Introduction

The adoption of predictive models within the industrial sector promises to bring substantially increased operational effectiveness as well as the development of new services and products [1]. With the increasing availability of low-cost sensors, it is possible to collect real-time vibration and audio data from critical locations inside automated manufacturing machines. However, converting raw sensor data into accurate and actionable information is still a challenge. While the adoption of automated manufacturing machines has led to more efficient manufacturing processes, there is still a need for improved monitoring solutions [2].

For milling and turning processes, it is common for a human operator to use sound, vibration or visual methods to identify whether the mechanical process is operating smoothly. Replicating such monitoring techniques with computer systems, however, has proven difficult because of the volume, sparsity and noise that is associated with sensor data [3]. With the adoption of Computer Numerical Control (CNC) machines and production-line manufacturing, manufacturing machines operate with less human input. However, many automated manufacturing machines are unable to detect when a process is faulty until damage has been caused to the product or the machine. There is a need for better automated techniques to monitor manufacturing machines and to ensure that they operate efficiently.

This work presents a technique for deducing the condition of a milling machine tool, using acceleration and audio signals. In particular, a pipeline for mapping raw sensor data to predicted tool condition values is developed. This work also presents solutions to several real-time challenges in automated monitoring of manufacturing systems, namely: high data volume, high noise content, and information sparsity [3]. In particular, the data volume challenge is alleviated by preprocessing the data into blocks. The data sparsity challenge is overcome by using several well-established signal processing techniques, including the Fast Fourier Transform (FFT). Finally, Gaussian process regression (GPR) is used to learn a mapping between the preprocessed sensor data and physical condition of the machine tool, in a manner that is not sensitive to noise in the observed condition.

Accurate monitoring of tool condition during machine operations may provide significant benefits in terms of cost and part quality. It is reported that 6.8 % [4] of the downtime in modern machine tools is attributed to tool failure, while one study puts the figure closer to 20 % [5]. There is significant theoretical study and background literature that deals with sensing, processing methods and reliability assessment of manufacturing systems [6–8]. A review of monitoring and control of manufacturing processes for increasing process efficiency, tooling life, and product quality is provided in [9]. Extensive research has been performed to establish methods for monitoring tool condition in milling operations. In these methods, various signals such as acoustic emission, machine vibration, cutting force, motor torque and motor current have been used to predict tool condition [10–16]. A detailed study of the correlation between tool wear and many of these signal types is presented in [17]. It has been reported that the ratio of high-band and low-band machine vibration is correlated with tool wear in some experiments [18]. A number of researchers have attempted to use the skew and kurtosis coefficients of the audio and acceleration time series to predict the condition of the tool, but with mixed results [16, 19, 20]. However, the

adoption of advanced monitoring techniques in the manufacturing domain remains slow, despite ongoing research in the field [21]. It is reported that development of algorithms and paradigms that are autonomous from machine tool operators is critical to the adoption of advanced monitoring techniques on the shop floor [21]. These autonomous algorithms should perform signal feature extraction and decision making with minimal intervention from the operator, who should provide only simple input and information [21–23].

In recent times, there has been increased interest in using machine learning techniques to develop accurate predictive models that perform well across a range of operating conditions. In [24], GPR is used successfully to predict tool condition, based on machine vibration and acoustic emission. The use of neural networks in tool condition prediction is also discussed extensively in the literature [23, 25–27]. It has been reported that neural networks can be used to simultaneously predict multiple values of interest, including tool condition, cutting force, and surface condition [28]. A wide range of tool condition monitoring techniques utilizing neural networks have also been reviewed by Dimla, Lister, and Leighton [22]. They concluded that artificial neural networks perform well on carefully selected experimental data, but there is a need for a multi-level system capable of handling unprocessed data. In particular, a useful tool condition monitoring system must automate the pre-processing, featurization and prediction steps, so the system can operate directly on raw sensor signals [22].

The remainder of this paper is organized as follows: In Section 2, the experimental setup and wireless data collection hardware is introduced. In Section 3, an overview of the data processing pipeline and predictive model is provided. In Section 4, several signal processing techniques are used to reduce the dimensionality and sparsity of the data. In Section 5, a GPR model is developed to predict the tool condition of the CNC machine. The model is trained on the experimental data and the results are presented in Section 6. Section 7, discusses how the predictive model can be represented using the standardized Predictive Model Markup Language (PMML). The paper is concluded with a summary and discussion.

2 Experimental Design and Monitoring Hardware

This section describes how time series acceleration and acoustic data are collected from a CNC milling machine, namely a Mori Seiki NVD1500DCG. As shown in Fig. 1, a waterproof sensor unit from Infinite Uptime is attached to the vise of the milling machine. The sensor unit is capable of measuring both the audio and triaxial acceleration signals inside the milling machine. The acceleration signal is recorded in the x-, y- and z-directions. The acceleration sampling rate must be high enough to capture the signal from the four tool flutes as they remove material from the part. An acceleration sample rate of 1000 Hz is chosen to capture the 200 Hz signal generated by the cutting tool when the spindle speed is set to 3000 RPM. The audio signal is recorded at 8000 Hz. Data is streamed from the sensor to a laptop computer using a Universal Serial Bus (USB) connection.

The machining data, such as tool position and rotation speed, are recorded from the FANUC controller. An MTConnect [29] agent is used to synchronize the data from the milling machine and stream it to a laptop computer, along with a timestamp. A post-processing step is used to

convert the machining data into a set of cutting operations performed by the machine. In the post-processing step, a simulation of the milling machine operations is used to distinguish actual material removal operations from other tool movements (e.g., movement above the workpiece) [30]. Using simulation, it is possible extract the behavior of the machine from the raw numerical control (NC) code. The details of the post-processing simulation step have been described in an earlier work [31].

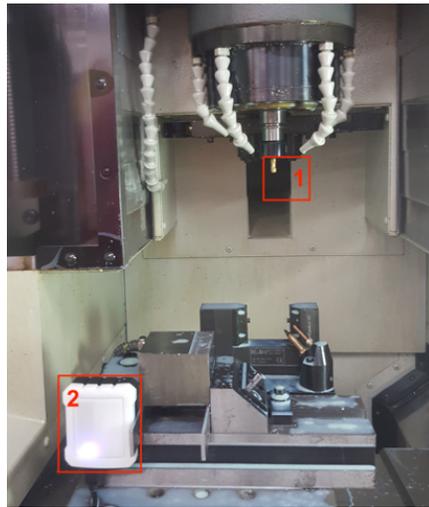


Figure 1. Experimental setup in a Mori Seiki NVD1500DCG milling machine showing (1) the cutting tool and (2) the sensor unit from Infinite Uptime.

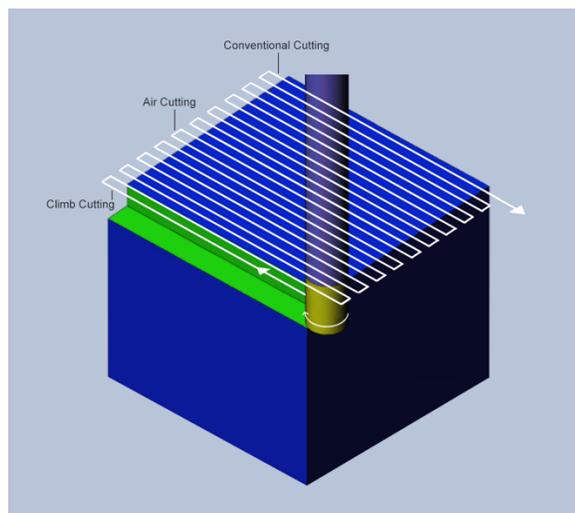


Figure 2. Machining strategy for tool condition experiments.

The milling machine was programmed to produce a number of simple parts by removing material from a square block of 1018 cold finish mild steel. The physical dimensions and mechanical properties of steel block are provided in Table 1. The trajectory path of the tool is

shown in Fig. 2. A total of 20 separate cutting operations were made in the production of each part. The milling machine was used to continuously produce parts until the cutting tool became severely damaged, or the cutting tool broke. The tool condition experiments were conducted using an Atrax solid carbide 4-flute square end mill, and a continuous supply of coolant. The properties of the end mill are listed in Table 2. The milling machine operating parameters and experimental statistics are shown in Table 3. Under standard operating conditions a feed rate of 152.1 mm/min and a spindle speed of 4500 RPM would be used. To accelerate tool wear, the feed rate was increased to 304.2 mm/min and the spindle speed was reduced to 3000 RPM. A cutting depth of 0.5mm was used. With these operating parameters, the life of the tool was reduced to 30 – 60 minutes. The predominant type of tool wear was abrasive flank wear, under both the normal and accelerated-wear operating conditions. However, rapid failure of the tool caused by chipping of the tool flank was occasionally observed under the accelerated-wear operating conditions. It should be noted that this type of failure is much more difficult to predict than gradual degradation.

A number of different methods have been proposed to measure the condition of a machine tool. Several authors investigating tool condition have used quantitative measurements, such as the wear depth, as tool condition measures [32]. However, these measures often fail to fully describe the tool condition, especially when there is irregular wear or chipping in the tool flank [25]. The condition of the milling machine tool is defined as, $y \in [0,1]$, based on the remaining lifetime of the tool, as estimated after manually examining the tool with a microscope. Tool condition estimation is based on the most heavily worn flute. The scale is defined such that 1.0 indicates a new tool in perfect condition, and 0.5 indicates the condition at which the tool would be replaced in a commercial manufacturing operation as judged by a machine operator. Fig. 3 shows examples of different levels of tool condition. A quantitative description of the proposed tool condition metric is provided in Table 4.

Table 1. Physical and mechanical properties of the 1018 cold finish mild steel block, used in the tool condition experiments.

Physical or Material Property	Value
Work piece material	1018 cold finish mild steel
Work piece dimensions	63.5mm x 63.5mm square, cut to a length of 56mm
Tensile strength	64,000 psi
Yield strength	54,000 psi
Brinell hardness	126

Table 2. Physical and material properties of the Atrax solid carbide square end-mill, used in the tool condition experiments

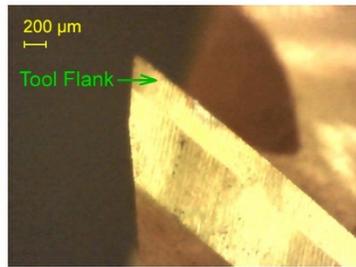
Physical or Material Property	Value
Tool Material	Solid Carbide
Finish/Coating	Titanium Nitride
Mill Diameter	3/16" (4.7625 mm)
Shank Diameter	3/16" (4.7625 mm)
Flute Type	Spiral
Number of Flutes	4

Table 3. Experimental statistics and operating parameters for the Mori Seiki NVD1500DCG milling machine

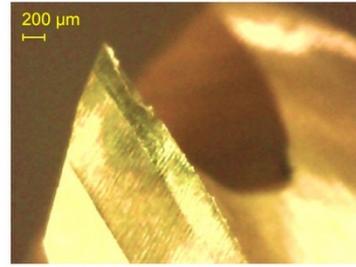
Parameter/Statistic	Value
Parts produced	79
Machine tools used	18
Machine feed rate	304.4 mm/minute
Machine tool rotation rate	3000 revolutions per minute (RPM)
Machine tool chip load	0.001 in/tooth (0.0254 mm/tooth)

Table 4. Quantitative tool condition evaluation rubric. Linear interpolation is used where the tool condition falls between two limits. The tool condition parameter y is directly related to maximum flank wear on the range $y \in [0.5, 1.0]$. For $y \in [0, 0.5)$ the evaluation of tool condition is judged on a range of possible failure modes.

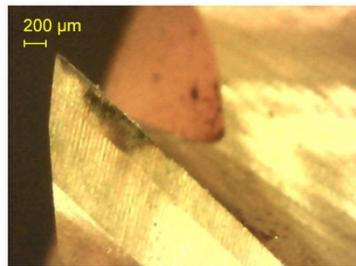
Tool Condition (y)	Value
1.0	New tool
0.5	Maximum flank wear of 50 microns on at least one of the four tool flutes.
0.3	Maximum flank wear of 150 microns on at least one of the four tool flutes, or mean flank wear of 100 microns on at least one of the four tool flutes.
0.2	Nose, flank or crater wear exceeding 200 microns on at least one of the tool flutes. Chipping is commonly observed at this wear level. Milling is not continued if this wear level is observed after the completion of a part.
0.1	Flank wear or chipping exceeding 300 microns. Damage to the tool is clearly visible without a microscope. Milling is stopped immediately if this wear level is reached.
0	Complete failure of the tool. Milling is stopped immediately if this wear level is reached.



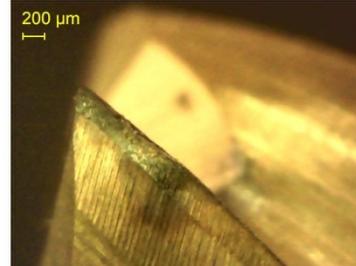
(a) $y = 0.9$



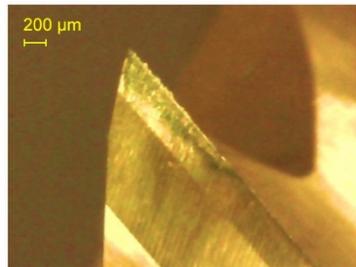
(e) $y = 0.5$



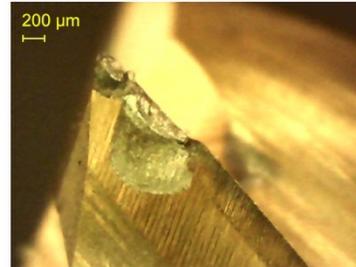
(b) $y = 0.8$



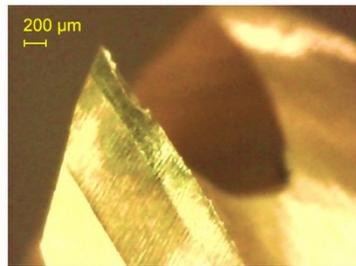
(f) $y = 0.4$



(c) $y = 0.7$



(g) $y = 0.2$



(d) $y = 0.6$



(h) $y = 0.0$

Figure 3. Solid carbide end mill flute in different states of condition. A lower value of y indicates higher tool wear condition.

3 A Pipeline for Time Series Data Processing

One main contribution of this work is to provide a data processing pipeline that can be used to generate intermittent predictions from a high-volume stream of noisy time series data. Standardized data processing techniques and model representations are used where possible, to ensure that the data processing pipeline can easily be deployed in a real manufacturing facility. In this section, an overview of the proposed data processing pipeline is provided. In subsequent sections, each component of the data processing pipeline is described in detail, using tool condition prediction as an example use-case.

Fig. 4 shows the proposed data processing pipeline. Acceleration and audio time series signals are recorded at the milling machine. In the first component of the pipeline, information from the CNC machine controller is used to label the blocks of time series data according to the machine action. This helps to organize the large time series signals into more manageable chunks. In the second component, the time series data is transformed into the frequency domain. This transformation allows the volume of the data to be reduced without significant loss of relevant information. The FFT is chosen to create a frequency domain representation of the raw time series data, as it is computationally efficient and widely understood. In the final component, a trained predictive model is used to map the frequency-domain data to the output value, namely the tool condition.

The predictive model is represented using the standardized Predictive Model Markup Language (PMML) format [33]. This allows the trained model to be easily transferred from research to the factory floor, in a controlled and standardized manner. A scalable and production-ready implementation of this pipeline could be composed of a preprocessing module and a scoring module. The preprocessing module would reduce the time series data to feature vectors using FFT, and the scoring module would generate predictions using a PMML-compliant scoring engine. This pipeline would be relatively easy to implement in any computing environment that supports FFT and PMML.

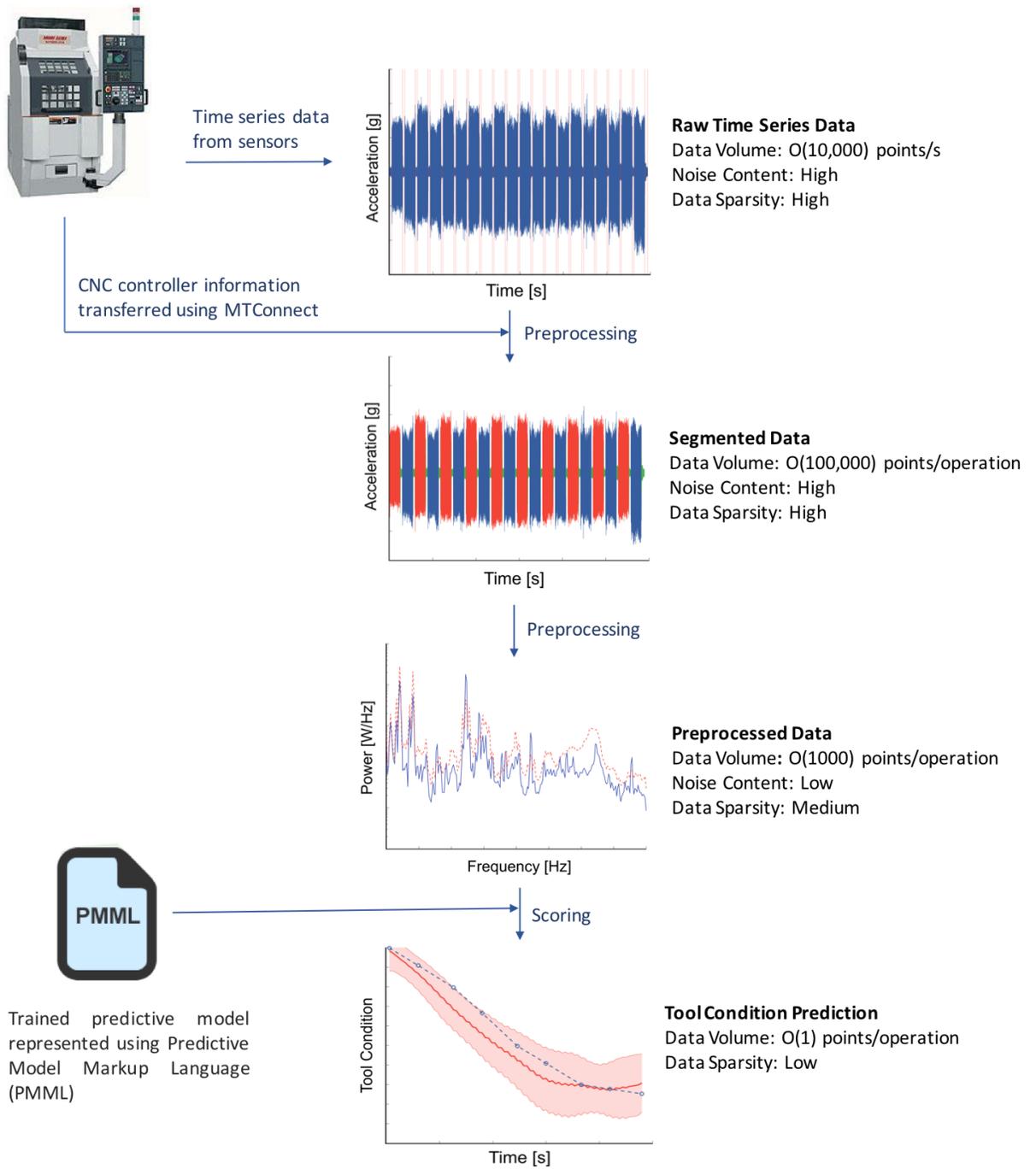


Figure 4. Diagram showing the proposed data processing pipeline. In the pipeline, raw time series data from the milling machine sensors is mapped to tool condition predictions.

4 Time Series Data Segmentation

In this section, the time series data is segmented into blocks and labelled using the controller data. The rate of data produced by the acceleration and audio sensors is such that directly learning from the data is difficult. Furthermore, the acceleration and audio data is delivered as a continuous stream of samples, while the actual manufacturing process is a series of more structured operations. The amplitude of the acceleration signal is related to the type of cutting operation the machine is performing, as can be seen in Fig 5. A human operator monitoring the manufacturing machine would be aware of this relationship, and use this supplementary information when analyzing the state of the machine. Therefore, segmenting the time series data according to machine operation is an important step for predicting tool condition.

The milling machine performs a number of different operations to produce a part. The data from the milling machine controller is used to automatically segment and label the time series data. Fig. 6 shows the time series data in the production of a part that involves 10 climb-cutting operations and 10 conventional-cutting operations. Each cutting operation is separated by a brief air-cutting operation. The term “air-cutting” is used to describe the operation of running the spindle without removing material. The terms “climb cutting” and “conventional cutting” refer to the relationship between the rotation direction to the feed direction. A detailed description of these operations is provided in an earlier work [31].

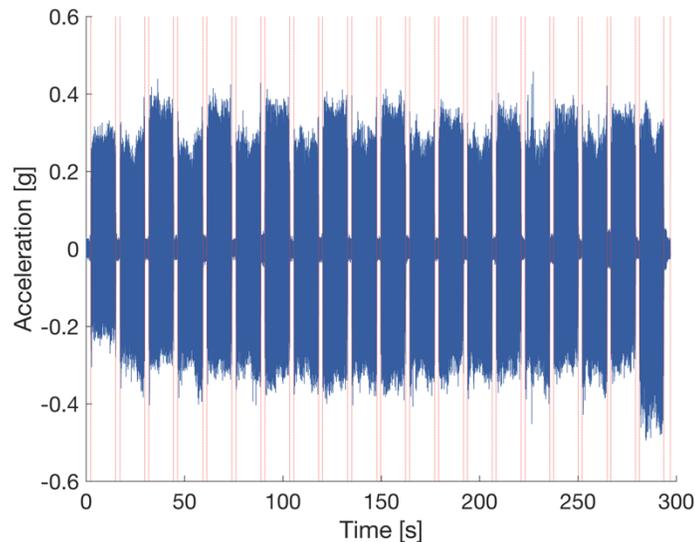


Figure 5. Acceleration time series measured while a single part was being produced. The time series data shows that the machine performs a series of repetitive actions, separated by brief idle periods.

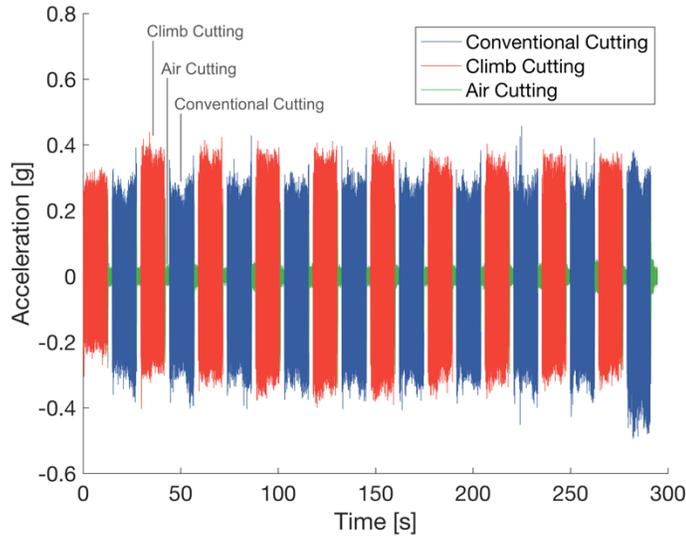


Figure 6. The acceleration time series after each action is labelled using data from the machine controller

4.1 AGGREGATION AND NOISE REDUCTION

As the audio and vibration signals are periodic, it seems natural to analyze the signals in the frequency domain. Technology involving filtering, modulation, and wave propagation often relies upon spectral analysis via, for example, the Fourier transform. A human machine operator relies on spectral analysis as well [34]. The machine operator classifies audio and vibration signals according to high and low pitch as well as purity, even if only subconsciously. This section describes how the sparse and noisy time-domain signals are transformed into intermediate representations which capture the sort of information required to evaluate the state of a manufacturing process.

The goal of this section is to obtain the power spectrum of the acceleration and audio signals, whilst reducing the influence of noise. It is assumed that the frequency content of the recorded signals is solely dependent on the condition of the machine tool and the type of operation being performed by the machine. It follows from this assumption that the frequency content of the signals should be relatively constant over the duration of each machine operation. It is also assumed that the audio and acceleration signals are laden with noise.

The discrete Fourier transform (DFT) is often used to estimate the power spectrum of a signal. However, the standard method of estimating the power spectrum using the Fourier transform tends to be susceptible to noise caused by imperfect and finite data [35]. An alternative technique, introduced by Welch [36] that involves averaging multiple periodograms, is used to overcome some of these problems. Welch's method is an improvement on the standard periodogram method, in that it reduces noise in the estimated power spectra in exchange for reducing the frequency resolution [37]. In Welch's method, a discrete time series signal s , is divided into K successive blocks s_m , using a window function w :

$$s_m(n) = w(n)s(n + mR), \quad n = 0 \dots M - 1, \quad m = 0 \dots K - 1 \quad (1)$$

where M is the length of the window and R is the window hop size [38]. The parameters M and R control the number of periodograms which are averaged to give the periodogram estimate. The standard periodogram method uses a rectangular window, but in this work the common Hann window function is used to reduce the spectral leakage [34]:

$$w(n) = \begin{cases} 0.5 \left(1 - \cos \left(\frac{2\pi n}{M-1} \right) \right) & \text{if } n \leq M-1; \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The periodogram of the m^{th} block is calculated using the Fourier transform:

$$\mathbf{p}_m(\omega_k) = \frac{1}{M} \left| \sum_{n=0}^{M-1} s_m(n) e^{-\frac{i2\pi nk}{N}} \right|^2. \quad (3)$$

where ω_k is the k^{th} point in the discretized frequency domain. The Welch estimate of the power spectral density is given by:

$$\hat{\mathbf{s}}(\omega_k) = \frac{1}{K} \sum_{m=0}^{K-1} \mathbf{p}_m(\omega_k). \quad (4)$$

Welch's method essentially computes the average of periodograms across time. Each periodogram is obtained from a windowed segment of the time series. In this study, a window overlap of 50% is used. A Hann window length of 256 points is chosen for the vibration and audio signals. The length of the window is equal to the number of intervals in the discretized frequency domain. In this study, discretizing the frequency domain over 256 intervals provides a minimal but sufficient representation of the true periodogram.

Alternative methods such as discrete wavelet transform (DWT) could be used to construct a time-frequency representation of the data in each time series block. However, the aim of this study is not to explore how the signal changes with time during a single machine operation. Instead, the aim is to explore how the average signal generated during one machine operation compares with the average signal generated during a previous machine operation. Therefore, applying DFT with Welch's method is simpler and more suitable in this context.

When observing a milling machine operates in a lab setting, the acceleration signal amplitude increases as the cutting tool wears. As shown in Fig. 7, the acceleration signal amplitude is much larger when the tool is worn, across most frequencies. Another interesting observation is that the sharp tool produces a more peaked periodogram, corresponding to pure vibration modes, while the worn tool produces a flatter periodogram. The audio periodogram recorded with a sharp tool also differs from that recorded with a worn tool, as can be seen in Fig. 8.

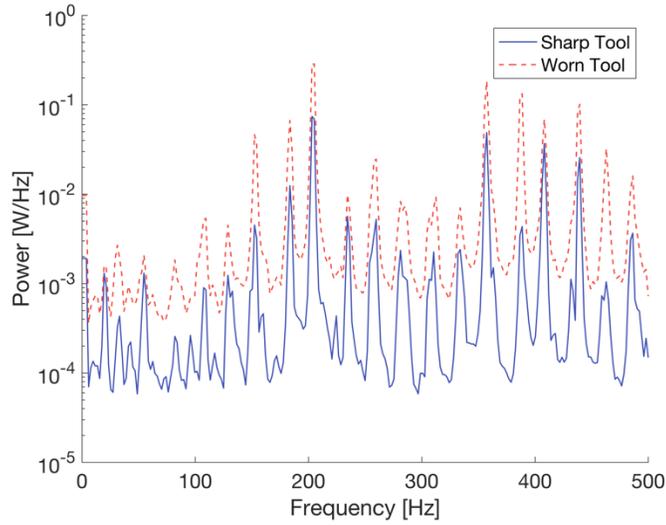


Figure 7. Comparison of the acceleration periodograms recorded with a sharp tool and a worn tool, while a climb-cutting operation was being performed

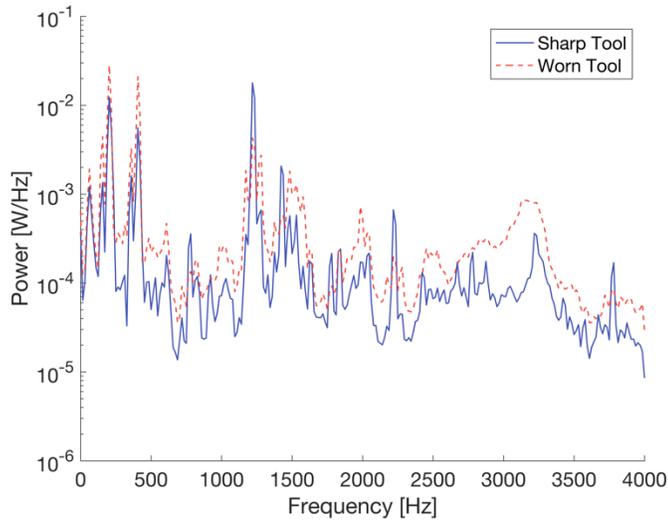


Figure 8. Comparison of the audio periodograms recorded with a sharp tool and a worn tool, while a climb-cutting operation was being performed

5 Tool Condition Prediction with Gaussian Process Regression

In this section, a mapping between the periodograms and the tool condition is developed using the GPR technique. The GPR technique provides a posterior distribution over the output, by comparing input features using a kernel function. The regression task has a dataset $\mathcal{D} = \{\mathbf{x}^i, y^i\}_{i=1}^n$ with D -dimensional inputs \mathbf{x}^i and scalar outputs y^i . The goal of GPR is to learn the unknown function $y^i = f(\mathbf{x}^i)$ by incorporating prior knowledge captured in the historical data.

In GPR, a Gaussian process (GP) is used as a prior to describe the distribution on the target function value $f(\mathbf{x})$ at an unseen input \mathbf{x} . A GP is a generalization of the Gaussian probability distribution for which any finite linear combination of samples has a joint Gaussian distribution [39]. A GP can be fully specified by its mean function $m(\mathbf{x})$ and covariance kernel function $k(\mathbf{x}, \mathbf{x}')$:

$$p(\mathbf{f}^{1:n}) = \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (5)$$

where $\mathbf{f}^{1:n} = (f(x^1), f(x^2), \dots, f(x^n))$ are latent function values. The mean function and covariance kernel are defined as [40]:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \quad (6)$$

$$k(\mathbf{x}, \mathbf{x}') = \text{cov}(f(\mathbf{x}), f(\mathbf{x}')). \quad (7)$$

The mean function $m(\mathbf{x})$ captures the overall trend in the target function value, and a kernel function $k(\mathbf{x}, \mathbf{x}')$ is used to approximate the covariance between the two function values $f(\mathbf{x})$ and $f(\mathbf{x}')$.

The acceleration and audio periodograms are used as input features to the tool condition model. The vibration periodogram $\hat{\mathbf{s}}_v^i \in \mathbb{R}^{256}$ and acoustic periodogram $\hat{\mathbf{s}}_a^i \in \mathbb{R}^{256}$ are defined for each milling machine operation i . The condition of the milling tool tends to change gradually under normal operation. For this reason, the previous condition of the milling tool is also included in the feature vector. Altogether, input features for each milling operation i , are denoted as:

$$\mathbf{x}^i = \begin{bmatrix} c^i \\ \hat{\mathbf{s}}_v^i \\ \hat{\mathbf{s}}_a^i \end{bmatrix}. \quad (8)$$

where c^i is the best estimate of the previous tool condition. Let the data from the production of each part be represented as the subset $\mathcal{D}_p = \{\mathbf{x}^k, \mathbf{y}^k\}_{k=1}^m$, where m is the number of points recorded over the duration of the experiment. During the training procedure, the previous tool condition is known, so c^k can be set equal to \mathbf{y}^{k-1} :

$$c_{training}^k = \begin{cases} 1 & \text{for } k = 1, \\ \mathbf{y}^{k-1} & \text{otherwise.} \end{cases} \quad (9)$$

To make a new prediction using the scoring procedure, the feature vector \mathbf{x}^{new} first needs to be derived from the milling machine data. In a manufacturing setting, the acceleration and audio periodograms can be calculated immediately after each operation is performed by the milling

machine. However, the previous tool condition value y^{k-1} , will not be known. Therefore, the previous tool condition prediction, \hat{y}^{k-1} , is used the scoring procedure. That is:

$$c_{testing}^k = \begin{cases} 1 & \text{for } k = 1, \\ \hat{y}^{k-1} & \text{otherwise.} \end{cases} \quad (10)$$

This makes the prediction process recursive. The first prediction is made by assuming a new tool, i.e. $c_{testing}^k = 1$. All subsequent predictions are made using the previous prediction as a starting point.

5.1 NOISE MODEL

Each tool condition label y^i in the training and testing datasets are assigned manually, so they are likely to contain some random error. This error is modelled as random noise in the output value. Each observed value is modelled by the function $y^i = f(\mathbf{x}^i) + \epsilon$ where ϵ is the noise term. It is assumed that this noise follows an independent, identically distributed Gaussian distribution with zero mean and variance σ_ϵ^2 :

$$\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2). \quad (11)$$

5.2 KERNEL FUNCTION

The covariance kernel function provides an efficient method to estimate the correlation between two feature vectors. In GPR, the kernel function is used to estimate the covariance between two input vectors, \mathbf{x}^i and \mathbf{x}^j . The type of kernel function chosen can strongly affect the representability of the GPR model, and influence the accuracy of the predictions. The *squared exponential* (SE) kernel is a common choice for a GPR model:

$$k_{SE}(\mathbf{x}^i, \mathbf{x}^j) = \sigma^2 \exp\left(-\frac{1}{2\ell^2} \|\mathbf{x}^i - \mathbf{x}^j\|^2\right), \quad (12)$$

where the kernel function is described by the hyperparameters, σ and ℓ . The hyperparameter ℓ is commonly referred to as the length scale. The length scale quantifies whether two points at a certain distance apart in the input space are considered close together [39]. The signal variance σ^2 quantifies the overall magnitude of the covariance value. While the squared exponential (SE) kernel function is a good choice for many applications, it does not allow the length scale to vary for each dimension in the feature vector. A common solution is to use an *automatic relevance determination* (ARD) kernel, which assigns a different length scale to each dimension [41]. The ARD SE kernel is commonly used with GPR:

$$k_{ARD\ SE}(\mathbf{x}^i, \mathbf{x}^j) = \sigma^2 \exp\left(-\frac{1}{2} (\mathbf{x}^i - \mathbf{x}^j)^T \text{diag}(\boldsymbol{\ell})^{-2} (\mathbf{x}^i - \mathbf{x}^j)\right). \quad (13)$$

An ARD SE kernel provides the flexibility to adjust the relevance (weight) of each parameter in the feature vector where the parameter vector $\boldsymbol{\ell} = (\ell_1, \dots, \ell_i, \dots, \ell_m)$ quantifies the relevance of the input features. The ARD SE kernel has been shown to have good performance on a number of tasks including robot arm control and tool condition prediction [39, 42]. However, when using the ARD SE kernel, the number of hyperparameters grows linearly with the dimensional size of the feature vector. This can make the model prone to overfitting, depending on the hyperparameter optimization strategy [41].

An ideal kernel for the tool condition model would allow the length scale of each periodogram to be varied independently, without introducing a large number of additional hyperparameters. One common way to build a kernel over multiple data types is to add the kernels together [43]. A new kernel is created by combining several Gaussian kernels. The resulting *sum of square exponential* (SSE) kernel is defined as follows:

$$k_{SSE}(\mathbf{x}^i, \mathbf{x}^j) = \sigma_1^2 \exp\left(-\frac{1}{2\ell_1^2} \|c^i - c^j\|^2\right) + \sigma_2^2 \exp\left(-\frac{1}{2\ell_2^2} \|\hat{\mathbf{s}}_v^i - \hat{\mathbf{s}}_v^j\|^2\right) + \sigma_3^2 \exp\left(-\frac{1}{2\ell_3^2} \|\hat{\mathbf{s}}_a^i - \hat{\mathbf{s}}_a^j\|^2\right), \quad (14)$$

where $\sigma_1, \sigma_2, \sigma_3, \ell_1, \ell_2$ and ℓ_3 are the parameters to be determined for the SSE kernel function. The first term in the SSE kernel captures the similarity of the previous tool condition, for two inputs \mathbf{x}^i and \mathbf{x}^j . The second and third terms capture the similarity of the acceleration and audio periodograms, respectively. It is common practice in GPR model to include the noise term σ_ϵ^2 as another hyperparameter to be optimized in the training procedure. Altogether, a model with the SSE kernel function is described by seven hyperparameters, $\boldsymbol{\theta} = [\sigma_1, \sigma_2, \sigma_3, \ell_1, \ell_2, \ell_3, \sigma_\epsilon]$. When compared to the ARD kernel, the SSE kernel has fewer hyperparameters, but still allows the length scale of the previous state and periodograms to be adjusted independently.

5.3 TRAINING PROCEDURE

It follows from the assumption of Gaussian noise, a posterior distribution over Gaussian process functions can be analytically inferred. Hence, it is possible to analytically derive the marginal likelihood of the observed function values \mathbf{y} , given only hyperparameters $\boldsymbol{\theta}$, and the input vectors. The marginal distribution of the observations can be expressed as [41]:

$$p(\mathbf{y}^{1:n} | \boldsymbol{\theta}) = \int p(\mathbf{y}^{1:n} | \mathbf{f}^{1:n}, \boldsymbol{\theta}) p(\mathbf{f}^{1:n} | \boldsymbol{\theta}) d\mathbf{f}^{1:n}. \quad (15)$$

The unknown function \mathbf{f} can be marginalized out of Eq 15 to obtain the marginal likelihood of the training observations. The hyperparameters $\boldsymbol{\theta}$ are chosen to maximize the marginal likelihood of the observations in the training dataset \mathcal{D} . An optimization equation is formed to maximize the marginal likelihood, and obtain the optimum hyperparameters $\boldsymbol{\theta}^*$ [41]:

$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} \log p(\mathbf{y}^{1:n} | \boldsymbol{\theta}), \quad (16)$$

$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} \left(-\frac{1}{2} (\mathbf{y}^{1:n})^T (\mathbf{K} + \sigma_{\epsilon}^2 \mathbf{I})^{-1} \mathbf{y}^{1:n} - \frac{1}{2} \log |\mathbf{K} + \sigma_{\epsilon}^2 \mathbf{I}| - \frac{n}{2} \log 2\pi \right) \quad (17)$$

Finding the optimum hyperparameters using Eq 17 requires an iterative approach, as the value of the kernel matrix \mathbf{K} is inherently dependent on the hyperparameters. The process for obtaining the optimum hyperparameters is well documented in the literature [41]. In this study, the MATLAB GPML (Gaussian Processes for Machine Learning) library [44] is chosen to optimize the hyperparameters.

5.4 SCORING PROCEDURE

In machine learning, the scoring process generally involves using a predictive model to estimate some output value based on a set of input features \mathbf{x} . In GPR, the aim of the scoring procedure is to return a posterior distribution f^{new} on the output value, based on the unseen input \mathbf{x}^{new} . In the case where the mean function is zero, the (hidden) response value f^{new} and the observed outputs $\mathbf{y}^{1:n} = \{y^1, \dots, y^n\}$ follow the joint Gaussian distribution as:

$$\begin{bmatrix} \mathbf{y}^{1:n} \\ f^{new} \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} (\mathbf{K} + \sigma_{\epsilon}^2 \mathbf{I}) & \mathbf{k} \\ \mathbf{k}^T & k(\mathbf{x}^{new}, \mathbf{x}^{new}) \end{bmatrix} \right), \quad (18)$$

where $\mathbf{k}^T = (k(\mathbf{x}^1, \mathbf{x}^{new}), \dots, k(\mathbf{x}^n, \mathbf{x}^{new}))$. The posterior distribution on the response f^{new} for the newly observed input \mathbf{x}^{new} given the historical data can then be expressed as a Gaussian distribution:

$$f^{new} \sim \mathcal{N}(\mu(\mathbf{x}^{new} | \mathcal{D}^n), \sigma^2(\mathbf{x}^{new} | \mathcal{D}^n)). \quad (19)$$

The posterior mean $\mu(\mathbf{x}^{new} | \mathcal{D}^n)$, and associated variance $\sigma^2(\mathbf{x}^{new} | \mathcal{D}^n)$, can be calculated directly. That is, the mean and the variance of the posterior distribution can be expressed as [41]:

$$\mu(\mathbf{x}^{new} | \mathcal{D}^n) = \mathbf{k}^T (\mathbf{K} + \sigma_{\epsilon}^2 \mathbf{I})^{-1} \mathbf{y}^{1:n}, \quad (20)$$

$$\sigma^2(\mathbf{x}^{new} | \mathcal{D}^n) = k(\mathbf{x}^{new}, \mathbf{x}^{new}) - \mathbf{k}^T (\mathbf{K} + \sigma_{\epsilon}^2 \mathbf{I})^{-1} \mathbf{k}. \quad (21)$$

As the posterior distribution is one-dimensional Gaussian, the posterior mean and variance are sufficient to fully describe the posterior distribution.

6 Results

When evaluating machine learning models, it is common practice to divide the data set into a training set and a testing set. The model is trained on the training set and then tested on the

previously unseen data in the testing set. In applied machine learning research, it is common to randomly assign data points to the training and testing sets, but this method is not recommended for time series data [45]. Instead, 3 experiments are selected for the testing set, and data from the remaining 15 experiments is assigned to the training set.

Two GP models are trained to predict tool condition; the first is trained with climb-cutting data from the training set, and the second is trained with conventional-cutting data from the training set. The two models are then used together to predict the condition of the tool for each test case. Predictions are made in the order that the testing data was recorded.

In our experiments, models are trained using three different kernel functions. The prediction accuracy for each case is shown in Table 5. A model is said to generalize well if it achieves a similar performance on the training and testing set, or overfit if it fits well to the training data set but not to the test data set. The results demonstrate that the model with the SE kernel generalizes well, but it underfits the training data. The model with the ARD SE kernel however overfits the training data, but the model does not achieve good performance on the testing set. The model with the SSE kernel generalizes well and achieves reasonably good performance.

Fig. 9 shows the model prediction results when the previous tool condition state c^i is included in the input features. On the other hand, Fig. 10 shows tool condition predictions made without the state feature. The predictions in Fig. 10 show much larger variation in time than those with the previous state feature, as seen in Fig. 9. This qualitatively indicates that the previous condition state is a useful feature in the model.

Fig. 9 and Fig. 10 also reveal that the confidence interval on the predicted value is large when the tool is worn. There are several reasons for this: firstly, the tool is more likely to break or undergo rapid condition change at this stage; hence, there is less certainty when predicting the condition of a heavily worn tool. Secondly, the training set contains less data for heavily worn tools, as some of the tests were ended abruptly when the tool broke. The reduced amount of training data for heavily worn tools makes future predictions less certain. Finally, the amplitudes of the acceleration signal generated by heavily worn tools have a much larger variance. For example, if a single flute on the tool is worn heavily, the tool is no longer symmetric so the corresponding acceleration signal will have a very high amplitude. On the other hand, if the tool is worn evenly the corresponding acceleration signal may have a much smaller amplitude. Many of these limitations may have been exacerbated by the accelerated nature of the tool wear experiment. It is also important to note that tool condition would rarely be allowed to pass below $y = 0.5$ in a real manufacturing setting. Once the tool exceeds this level of wear, the surface quality of the part is likely to deteriorate quickly.

Table 5. Model performance on training set and testing set. Performance is measured using root mean squared error (RMSE). A smaller value of RMSE indicates a better fit.

Model	RMSE	
	Training Set	Testing Set
GP with squared exponential (SE) kernel	0.102	0.151
GP with ARD squared exponential (ARD SE) kernel	0.011	0.191
GP with sum of squared exponential (SSE) kernel	0.021	0.063

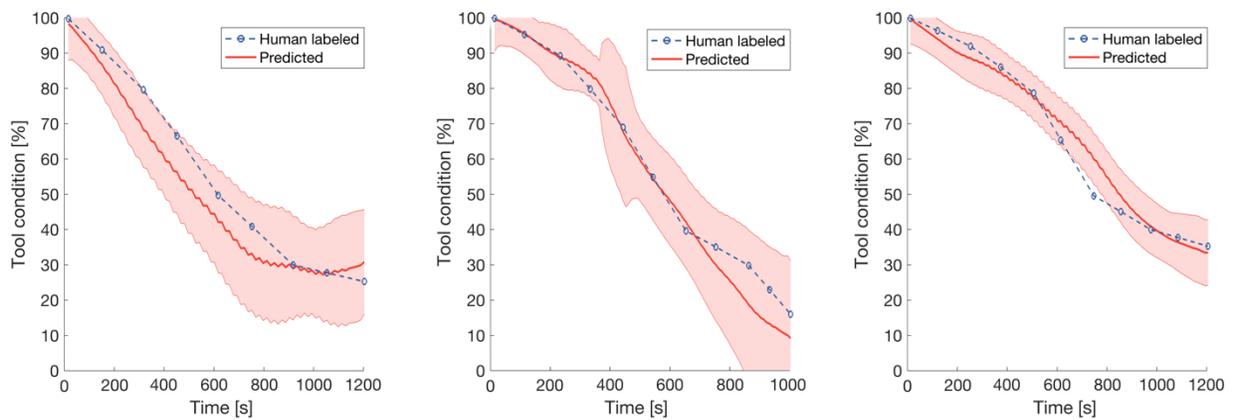


Figure 9. Tool condition prediction for three testing data sets using the model with the SSE kernel. Each plot represents a test where the milling machine was run until the cutting tool became severely worn or broken. The shaded region represents the 90% confidence interval for each prediction.

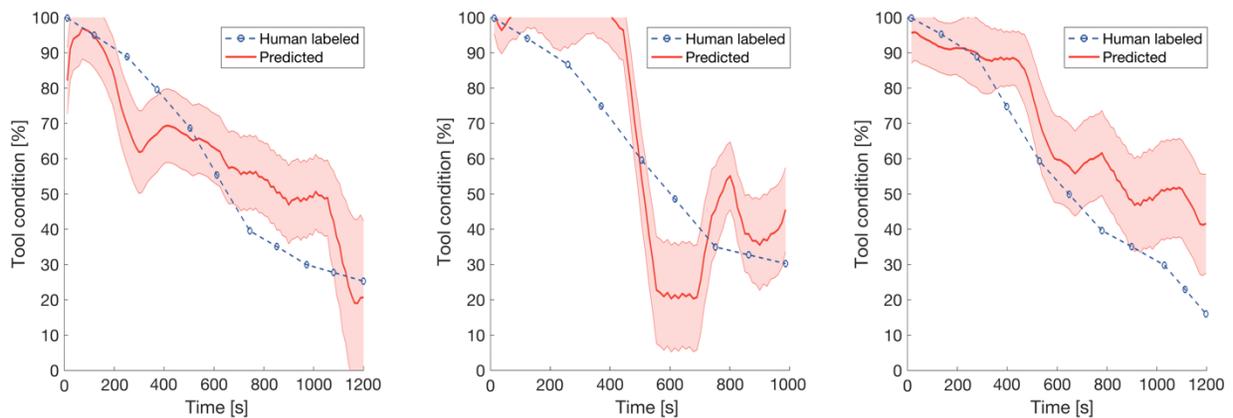


Figure 10. Tool condition prediction for three testing data sets using the model with the SSE kernel. For these plots, the previous state c^t was omitted from the feature vector when training and testing the model. The shaded region represents the 90% confidence

7 Standardizing Components of the Data Processing Pipeline

When developing a predictive model for industrial use, it is highly beneficial if the trained model can be represented in a standardized manner [46]. The need for standardization is particularly high in large organizations, where the predictive models may be shared between people or departments. This section discusses how the proposed data processing pipeline can be expressed in a standardized manner, using the PMML format [47].

The core of the proposed data processing pipeline is the GPR predictive model. The output of a trained GPR model is dependent on multiple factors, including the choice of mean kernel, the choice of covariance kernel, the model hyperparameters, and the training dataset. To transfer a trained GPR model between computing environments, it is also necessary to transfer this information. The PMML standard makes this information transfer less troublesome, by defining a standardized way to represent certain types of predictive models. A standardized representation for GPR models was introduced in the latest PMML specification, PMML 4.3 [48].

The GPR models developed in this work are expressed using PMML 4.3, where possible. The current PMML standard supports four covariance kernel types, namely the SE kernel, the ARD SE kernel, the absolute exponential kernel and the generalized exponential kernel [47]. Therefore, it is possible to represent the SE and ARD SE tool condition models with PMML 4.3. The PMML representation of a SE GPR model is shown in Fig. 11. The PMML representation of the ARD SE models is similar to that for the SE model, but it specifies a different covariance kernel and a different set of hyperparameters. The PMML 4.3 standard does not support composite covariance kernels that are created by the addition or multiplication of other standard kernel types. Unfortunately, the proposed SSE kernel cannot be represented using PMML 4.3. Future versions of PMML should consider supporting the addition and multiplication of covariance kernels, as addition or multiplication of kernels is commonly used to create new kernels with a range desirable properties [43]. A `CompositeKernel` element is proposed for consideration in future PMML releases. An example of the proposed `CompositeKernel` is shown in Fig. 12. The `operation` attribute of the `CompositeKernel` element is used to specify whether nested kernels are combined using addition or multiplication. The adoption of the proposed `CompositeKernel` would allow composite covariance kernels to be represented using PMML, in a fully standardized manner, with a minimal change to the existing specification.

```

<?xml version="1.0" encoding="utf-8"?>
<PMML xmlns="http://www.dmg.org/PMML-4_3" version="4.3">
  <Header copyright="dmg.org" />
  <DataDictionary numberOfFields="516">
    <DataField dataType="double" name="x1" optype="continuous"/>
    <DataField dataType="double" name="x2" optype="continuous"/>
    <DataField dataType="double" name="x3" optype="continuous"/>
    <DataField dataType="double" name="x4" optype="continuous"/>
    <DataField dataType="double" name="x5" optype="continuous"/>
    <DataField dataType="double" name="y1" optype="continuous"/>
  </DataDictionary>
  <GaussianProcessModel functionName="regression" modelName="Gaussian Process Model">
    <MiningSchema>
      <MiningField name="x1" usageType="active"/>
      ...
      <MiningField name="y1" usageType="predicted"/>
    </MiningSchema>
    <Output>
      <OutputField dataType="double" feature="predictedValue" name="MeanValue" optype="continuous"/>
      <OutputField dataType="double" feature="predictedValue" name="StandardDeviation" optype="continuous"/>
    </Output>
    <SquaredExponentialKernel gamma="20.6643" noiseVariance="0.48149">
      <Lambda>
        <Array n="1" type="real">0.114819</Array>
      </Lambda>
    </SquaredExponentialKernel>
    <TrainingInstances fieldCount="516" isTransformed="false" recordCount="435">
      <InstanceFields>
        <InstanceField column="x1" field="x1"/>
        ...
        <InstanceField column="x515" field="x515"/>
        <InstanceField column="y1" field="y1"/>
      </InstanceFields>
      <InlineTable>
        <Row>
          <x1>0.0016105</x1>
          ...
          <x515>1.00000</x515>
          <y1>0.168998</y1>
        </Row>
        ...
      </InlineTable>
    </TrainingInstances>
  </GaussianProcessModel>
</PMML>

```

Figure 11. Predictive model markup language (PMML) representation of the Gaussian process regression (GPR) model use to predict the condition of machine tools when performing climb-cutting operations. In this model the squared exponential (SE) covariance kernel is used.

```

<CompositeKernel function="add">
  <SquaredExponentialKernel gamma="20.6643" noiseVariance="0.48149">
    <Lambda>
      <Array n="1" type="real">0.114819</Array>
    </Lambda>
  </SquaredExponentialKernel>
  <SquaredExponentialKernel gamma="16.9983" noiseVariance="0.34189">
    <Lambda>
      <Array n="1" type="real">0.235221</Array>
    </Lambda>
  </SquaredExponentialKernel>
  <SquaredExponentialKernel gamma="22.1176" noiseVariance="0.58649">
    <Lambda>
      <Array n="1" type="real">0.212413</Array>
    </Lambda>
  </SquaredExponentialKernel>
</CompositeKernel>

```

Figure 12. An example of the proposed CompositeKernel element, expressed in the predictive model markup language (PMML). In this example, three squared exponential kernels are added together to form a composite covariance kernel.

8 Summary and Discussion

This study demonstrates how a data processing pipeline can be used to map raw sensor data to tool condition predictions. Information from the milling machine controller is used to divide the time series signals into discrete blocks of data. Each block of data is preprocessed using well-known and computationally efficient signal processing techniques. A non-parametric regression model, namely GPR, is then used to approximate the complex relationship between the preprocessed data and the target value. This pipeline-based approach is practical for use in a real manufacturing setting, as it allows raw sensor data to be mapped to tool condition predictions in a fast and robust manner.

The GP model provides confidence bounds for the predictive estimations, which are useful when interpreting the reliability of a prediction at some arbitrary time. The development of a tool condition model that produces a distribution over the current state of the machine is relatively novel, as most existing approaches just aim to predict a single value. The confidence bounds would likely prove useful in a practical application where the tool condition predictions are used to determine when to change machine tools.

The use of Welch’s method to calculate the power spectrum results in a noticeably smoother spectrum, with much less noise between the frequency peaks. Welch’s method also provides a way to reduce the time series from an arbitrary length to specific number of points. The transformation from the time domain to the frequency domain results in a significant dimensional reduction. The acceleration time series signal for each cutting operation contains on the order of 10,000 points and the audio time series contains on the order of 100,000 points. By applying Welch’s method, these signals are each reduced to 256 discrete points in the frequency domain. This corresponds to an order of magnitude dimension reduction in the acceleration data, and two orders of magnitude reduction in the acoustic data. Whilst the loss of frequency resolution is often undesirable in signal processing applications, it provides a convenient way to reduce the dimensionality of the data in this study.

Future work could investigate how the performance of the models with SE or ARD SE kernels can be improved. The model with the SE kernel did not fit the dataset well, as a single length scale is used for all dimensions in the feature vector. It is possible that the performance of the SE kernel model could be improved by applying a normalization technique to the feature vectors. The ARD kernel over-fitted the training dataset leading to bad performance on the testing set. The overfitting problem could be solved in several ways, such as adjusting the hyperparameter optimization strategy, using cross-validation, or applying a prior to the hyperparameters [41]. Future work could investigate how a model with an ARD SE kernel could be used to automatically determine the relevance of different audio and acceleration frequencies.

In this work, a composite covariance kernel is developed by combining a number of kernels. The introduction of the composite kernel allows the number of model hyperparameters to be reduced and improves model generalization for this application. Additionally, a CompositeKernel element is proposed as a potential extension of the current PMML specification. Future work could formalize the CompositeKernel element for adoption in the official PMML specification.

For the proposed data processing pipeline to have practical applications it must generalize well to a range of different machines and machine operations. Separate models for climb-cutting and conventional-cutting operations are trained in this work. However, it could be useful to train a single model that generalized well across multiple different machine operations. Future work could also investigate how to train a model which generalizes well across multiple types of milling machines. Another interesting research direction involves using a real-time adaptive GPR learning algorithm. In this case, the GPR algorithm only retains a subset of the observed data points. Using this technique would reduce computational demands and memory requirements, while improving generalizability across different parameter spaces [49].

Acknowledgements

The authors acknowledge the support by the Smart Manufacturing Systems Design and Analysis Program at the National Institute of Standards and Technology (NIST), US Department of Commerce. This work was performed under the financial assistance award (NIST Cooperative Agreement 70NANB17H031) to Stanford University. The authors would like to acknowledge the collaboration with the late Professor David Dornfeld of University of California Berkeley, who was instrumental in making the experiments possible.

Disclaimer

Certain commercial systems are identified in this paper. Such identification does not imply recommendation or endorsement by NIST; nor does it imply that the products identified are necessarily the best available for the purpose. Further, any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NIST or any other supporting U.S. government or corporate organizations.

References

- [1] M. Hermann, T. Pentek, and B. Otto, “Design Principles for Industrie 4.0 Scenarios,” in *49th Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 3928–3937.
- [2] Y. Altintas, *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*. Cambridge University Press, 2012.
- [3] J. Fan, F. Han, and H. Liu, “Challenges of Big Data Analysis,” *National Science Review*, vol. 1, no. 2, pp. 293–314, 2014.
- [4] G. Chryssolouris and M. Guillot, “A Comparison of Statistical and AI Approaches to the Selection of Process Parameters in Intelligent Machining,” *Journal of Engineering for Industry*, vol. 112, no. 2, pp. 122–131, 1990.
- [5] S. Kurada and C. Bradley, “A Review of Machine Vision Sensors for Tool Condition Monitoring,” *Computers in Industry*, vol. 34, no. 1, pp. 55–72, 1997.

- [6] G. Chryssolouris, *Manufacturing Systems: Theory and Practice*. Springer Science & Business Media, 2013.
- [7] P. Wright and E. Trent, “Metallurgical Appraisal of Wear Mechanisms and Processes on High-Speed-Steel Cutting Tools,” *Metals Technology*, vol. 1, no. 1, pp. 13–23, 1974.
- [8] S. Y. Tarasov, V. Rubtsov, and E. Kolubaev, “A Proposed Diffusion-controlled Wear Mechanism of Alloy Steel Friction Stir Welding (FSW) Tools Used on an Aluminum Alloy,” *Wear*, vol. 318, no. 1–2, pp. 130–134, 2014.
- [9] P. Stavropoulos, D. Chantzis, C. Doukas, A. Papacharalampopoulos, and G. Chryssolouris, “Monitoring and Control of Manufacturing Processes: A Review,” *Procedia CIRP*, vol. 8, pp. 421–425, 2013.
- [10] K. Chung and A. Geddam, “A Multi-sensor Approach to the Monitoring of End Milling Operations,” *Journal of Materials Processing Technology*, vol. 139, no. 1–3, pp. 15–20, 2003.
- [11] K. Jemielniak and P. Arrazola, “Application of AE and Cutting Force Signals in Tool Condition Monitoring in Micro-Milling,” *CIRP Journal of Manufacturing Science and Technology*, vol. 1, no. 2, pp. 97–102, 2008.
- [12] I. Inasaki, “Application of Acoustic Emission Sensor for Monitoring Machining Processes,” *Ultrasonics*, vol. 36, no. 1–5, pp. 273–281, 1998.
- [13] P. Stavropoulos, K. Salonitis, A. Stournaras, J. Pandremenos, J. Paralikas, and G. Chryssolouris, “Advances and Challenges for Tool Condition Monitoring in Micro-Milling,” in *Proceedings of the IFAC Workshop on Manufacturing Modelling, Management and Control, Budapest, Hungary, November, 2007*, pp. 157–162.
- [14] M. Shiraishi, “Scope of In-process Measurement, Monitoring and Control Techniques in Machining Processes — Part 1: In-process Techniques for Tools,” *Precision Engineering*, vol. 10, no. 4, pp. 179–189, 1988.
- [15] E. Kannatey-Asibu and D. A. Dornfeld, “A Study of Tool Wear Using Statistical Analysis of Metal-cutting Acoustic Emission,” *Wear*, vol. 76, no. 2, pp. 247–261, 1982.
- [16] R. G. Silva, R. L. Reuben, K. J. Baker, and S. J. Wilcox, “Tool Wear Monitoring of Turning Operations by Neural Network and Expert System Classification of a Feature Set Generated from Multiple Sensors,” *Mechanical Systems and Signal Processing*, vol. 12, no. 2, pp. 319–332, 1998.
- [17] P. Stavropoulos, A. Papacharalampopoulos, E. Vasiliadis, and G. Chryssolouris, “Tool wear predictability estimation in milling based on multi-sensorial data,” *The International Journal of Advanced Manufacturing Technology*, vol. 82, no. 1–4, pp. 509–521, 2016.

- [18] C. Doukas, P. Stavropoulos, A. Papacharalampopoulos, P. Foteinopoulos, E. Vasiliadis, and G. Chryssolouris, “On the Estimation of Tool-wear for Milling Operations based on Multi-Sensorial Data,” *Procedia CIRP*, vol. 8, pp. 415–420, 2013.
- [19] A. E. Diniz, J. J. Liu, and D. A. Dornfeld, “Correlating Tool Life, Tool Wear and Surface Roughness by Monitoring Acoustic Emission in Finish Turning,” *Wear*, vol. 152, no. 2, pp. 395–407, 1992.
- [20] D. E. Dimla, “Sensor Signals for Tool-wear Monitoring in Metal Cutting Operations—a Review of Methods,” *International Journal of Machine Tools and Manufacture*, vol. 40, no. 8, pp. 1073–1098, 2000.
- [21] R. Teti, K. Jemielniak, G. O’Donnell, and D. Dornfeld, “Advanced Monitoring of Machining Operations,” *CIRP Annals-Manufacturing Technology*, vol. 59, no. 2, pp. 717–739, 2010.
- [22] D. E. Dimla, P. M. Lister, and N. J. Leighton, “Neural Network Solutions to the Tool Condition Monitoring Problem in Metal Cutting—a Critical Review of Methods,” *International Journal of Machine Tools and Manufacture*, vol. 37, no. 9, pp. 1219–1241, 1997.
- [23] D. A. Dornfeld and M. DeVries, “Neural Network Sensor Fusion for Tool Condition Monitoring,” *CIRP Annals-Manufacturing Technology*, vol. 39, no. 1, pp. 101–105, 1990.
- [24] M. Ferguson, K. H. Law, R. Bhinge, and Y.-T. T. Lee, “A Generalized Method for Featurization of Manufacturing Signals, With Application to Tool Condition Monitoring,” in *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2017, p. V001T02A077–V001T02A077.
- [25] D. E. Dimla and P. M. Lister, “On-line Metal Cutting Tool Condition Monitoring. I: Force and Vibration Analyses,” *International Journal of Machine Tools and Manufacture*, vol. 40, no. 5, pp. 739–768, 2000.
- [26] N. Ghosh *et al.*, “Estimation of Tool Wear during CNC Milling using Neural Network-based Sensor Fusion,” *Mechanical Systems and Signal Processing*, vol. 21, no. 1, pp. 466–479, 2007.
- [27] T. Özel and Y. Karpat, “Predictive Modeling of Surface Roughness and Tool Wear in Hard Turning using Regression and Neural Networks,” *International Journal of Machine Tools and Manufacture*, vol. 45, no. 4–5, pp. 467–479, 2005.
- [28] E. Kilickap, A. Yardimeden, and Y. H. Çelik, “Mathematical Modelling and Optimization of Cutting Force, Tool Wear and Surface Roughness by Using Artificial Neural Network and Response Surface Methodology in Milling of Ti-6242S,” *Applied Sciences*, vol. 7, no. 10, p. 1064, 2017.
- [29] W. Sobel, “MTConnect Standard. Part 1 — Overview and Protocol,” Version 1.3.0, Jun. 2015.

- [30] R. Bhinge, J. Park, K. H. Law, D. A. Dornfeld, M. Helu, and S. Rachuri, "Toward a Generalized Energy Prediction Model for Machine Tools," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 4, p. 041013, 2017.
- [31] R. Bhinge *et al.*, "An Intelligent Machine Monitoring System for Energy Prediction Using a Gaussian Process Regression," in *2014 IEEE International Conference on Big Data*, 2014, pp. 978–986.
- [32] J. Yuan, J. M. Boyd, D. Covelli, T. Arif, G. S. Fox-Rabinovich, and S. C. Veldhuis, "Influence of Workpiece Material on Tool Wear Performance and Tribofilm Formation in Machining Hardened Steel," *Lubricants*, vol. 4, no. 2, p. 10, 2016.
- [33] A. Guazzelli, M. Zeller, W. Lin, and G. Williams, "PMML: An Open Standard for Sharing Models," *The R Journal*, vol. 1, no. 1, pp. 60–65, 2009.
- [34] P. Stoica and R. L. Moses, *Spectral Analysis of Signals*, vol. 452. Pearson Prentice Hall Upper Saddle River, NJ, 2005.
- [35] R. L. Allen and D. Mills, *Signal Analysis: Time, Frequency, Scale, and Structure*. John Wiley & Sons, 2004.
- [36] P. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, Jun. 1967.
- [37] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. Wiley India Pvt. Limited, 2009.
- [38] J. O. Smith III, *Spectral Audio Signal Processing*. W3K Publishing, 2011.
- [39] C. K. Williams and C. E. Rasmussen, "Gaussian Processes for Regression," *Advances in Neural Information Processing Systems*, pp. 514–520, 1996.
- [40] A. G. Wilson and R. P. Adams, "Gaussian Process Kernels for Pattern Discovery and Extrapolation," in *ICML (3)*, 2013, pp. 1067–1075.
- [41] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. 2006.
- [42] M. Ferguson, K. H. Law, R. Bhinge, D. Dornfeld, J. Park, and Y. T. Lee, "Evaluation of a PMML-based GPR Scoring Engine on a Cloud Platform and Microcomputer Board for Smart Manufacturing," in *2016 IEEE International Conference on Big Data*, 2016, pp. 2014–2023.
- [43] D. Duvenaud, "Automatic Model Construction with Gaussian Processes," University of Cambridge, 2014.

- [44] C. E. Rasmussen and H. Nickisch, “Documentation for GPML Matlab Code, Version 4.1,” 27-Nov-2017. [Online]. Available: <http://www.gaussianprocess.org/gpml/code/matlab/doc/>. [Accessed: 18-Jan-2018].
- [45] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, vol. 7. Cambridge University Press, 2004.
- [46] M. Zeller *et al.*, “Open Standards and Cloud Computing: KDD-2009 Panel Report,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 11–18.
- [47] The Data Mining Group, “The Predictive Model Markup Language, Version 4.3.” 01-Aug-2016.
- [48] J. Park *et al.*, “Gaussian Process Regression (GPR) Representation Using Predictive Model Markup Language (PMML),” *Smart and Sustainable Manufacturing Systems. Manuscript submitted for publication*, vol. 1, no. 1, p. 121, 2017.
- [49] J. Quiñonero-Candela and C. E. Rasmussen, “A Unifying View of Sparse Approximate Gaussian Process Regression,” *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.