ONLINE TOOL-WEAR PREDICTION USING GAUSSIAN PROCESS REGRESSION WITH BLOCKWISE ANALYTICS

Max Ferguson, Raunak Bhinge, Jinkyoo Park, Kincho Law

Abstract

This paper describes how the condition of a milling machine tool was predicted using vibration and audio signals measured near the milling machine blade. We demonstrate that the vibration and audio signals contain sufficient information to predict the milling tool condition. Sensor data is aggregated into blocks that correspond to the individual actions of the CNC milling machine. We demonstrate that this block-wise analysis technique allows advanced machine learning models to be applied at near real-time speed without sacrificing accuracy. The tool-condition model is shown to be very accurate, especially when predicting the condition of very worn tools.

Keywords – Machine learning, Gaussian Process Regression, Tool condition monitoring

1. INTRODUCTION

Modern wireless sensors are able to capture data at a high sample rate. However, the cost associated with sending this data to a central database can quickly exceed the cost of the device (ref). Furthermore, the storage cost of can also become prohibitively large, with a single day of uncompressed audio data requiring over 5 GB of storage capacity (ref). Several researchers have demonstrated that performing data aggregation on a wireless sensor can dramatically reduce the power consumption and bandwidth requirements without significant loss of information (Krishnamachari, B; ref). In this paper we demonstrate that the controller data from a CNC milling machine can be used to aggregate real-time sensor data, drastically reducing bandwidth requirements, whilst retaining important information about the operating condition of the machine. We demonstrate that the aggregated data contains sufficient information to predict the condition of the milling machine tool at near real-time.

Reliable tool-condition monitoring can provide a number of benefits for the manufacturing industry, like improved product quality and the prevention of tool breakage. 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. Previous researchers have demonstrated that the condition of the machine tool is can be inferred from features of the vibration and audio time series. (Reference) identified that there is a correlation between the tool-condition and the kurtosis coefficient of the audio time-series and the condition of the tool. In this paper we identify a number of features that can be used to predict the condition of a CNC milling machine tool. These features are calculated on the device and transferred to a cloud database, via a wireless connection. A Gaussian Process (GP) model is trained to predict the tool condition based on these features. Once trained, the GP model is used to predict the tool-condition in near real-time.

When aggregating time series data, it is common to discretized the time domain into finite blocks. In methods such as short-time Fourier transform (STFT), the time series is frequency content of the signal is calculated over local sections of the signal. In practice, this usually involves dividing a longer time-series signal into shorter segments of equal length and then computing the Fourier transform separately on each shorter segment. Whilst methods such as STFT are an effective way to monitor the transient nature of the frequency content, they suffer from the averaging effect; if the frequency content of the signal changes significantly during the STFT window, then the resultant STFT will represent a weighted average of the former and latter signals. This is highly undesirable, as the resultant SFTT does not represent the actual state of the system during the operation of the machine. To overcome this issue, we use the controller data to dynamically align the sampling windows with the actions performed by the machine.

The paper is organized as follows; In section 2, we explain the experimental setup and wireless data collection hardware. In section 3, we describe how the data is aggregated on the wireless sensor, and transmitted to a cloud server. In section 4 we describe how a Gaussian process regression model is trained to predict the tool condition of the CNC machine using the featurized data. Section 5 describes the results, and the paper is concluded with a brief summary and discussion.

2. EXPERIMENTAL DESIGN AND MONITORING HARDWARE

In this section we describe the experimental setup for collecting training and testing data. A number of simple parts were produced with a Computer Numerical Control (CNC) milling machine. As the parts were produced the condition of the milling machine blade deteriorated. The acceleration and acoustic signals inside the milling machine were measured throughout the duration of the milling process.

2.1 Experimental Setup

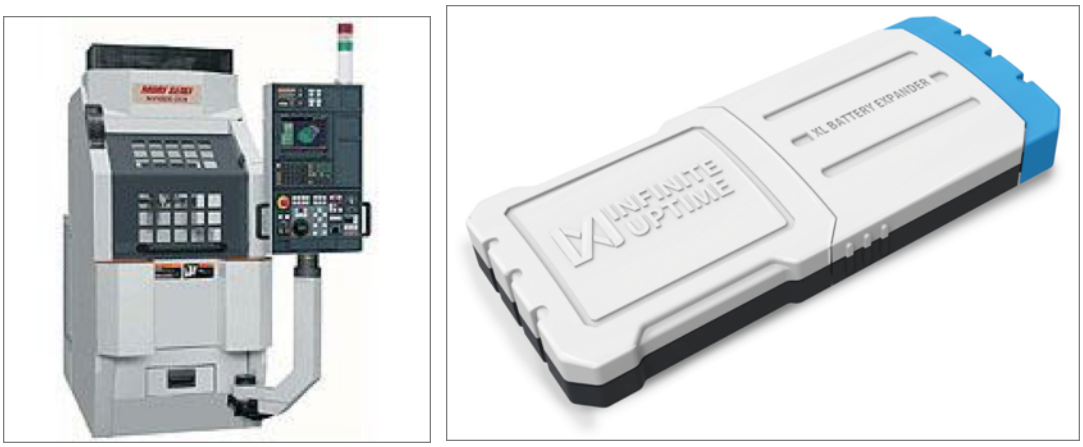


Figure 1. A Mori Seiki NVD1500DCG milling machine (left) and an InfiniteUptime sensor (right)

A Mori Seiki NVD1500DCG milling machine, similar to the one shown in Figure 1a, was programmed to produce a number of simple 'parts' by removing material from a solid steel block. Each part consisted of 20 separate cutting actions performed by the milling machine. On average, each cutting action was performed in about 3 seconds, and each part took about 1 minute to produce. The machine was instructed to produce parts until the cutting tool became severely damaged, or the cutting tool broke. A total of 23 tools were used to produce 100 parts.

The operating parameters of the machine were adjusted to artificially increase the rate of tool-wear. In a normal manufacturing environment, machine tools tend to last several days. In this experiment the operating lifetime of the machine tool was reduced to about 10 minutes by increasing the feed rate and the reducing the rotation speed.

An InfiniteUptime sensor was used to measure the audio and acceleration signal inside the milling machine while the machine was operational. The acceleration signal was recorded at 1000 Hz while the audio signal was recorded at 8000 Hz. The acceleration was measured in three axes. The sensor was waterproof, allowing it to be placed directly beneath the part being manufactured.

3. DATA AGGREGATION

Recording the time series

The acceleration and acoustic signals are measured continuously during the operation of the machine. We denote the three acceleration time-series signals as $s_x(t)$, $s_y(t)$, $s_z(t)$ and the audio signal as $s_a(t)$. The acceleration signals are recorded with a 1000 Hz sample rate and the audio signals are recorded with an 8000 Hz sample rate. Initial investigation revealed that the direction of the measured acceleration is not a useful factor in prediction of tool condition, so the acceleration time series signals are combined:

$$s_v(t) = \sqrt{s_x(t) + s_y(t) + s_z(t)}$$

Where $s_v(t)$ is the magnitude of acceleration measured at the device. We use the symbol $s(t)$ to refer to the time series signals collectively.

Defining actions and action types

The milling machine performs a number of operations to produce a part. We classify each operation as either a climb cutting, conventional cutting or air cutting operation, based on the type of cutting strategy that the machine is employing whilst performing an operation on the part. A physical interpretation of these actions is provided in [ref]. We use the superscript $i \in 1 \ldots n$ to denote operation performed by the machine, and the superscript $j \in 1,2,3$ to denote the type of action (air cutting, conventional cutting, climb cutting). The time type of action being performed by the machine at any time, t, can be obtained from the G-code provided by the milling machine [Raunak to check]. Figure 2 demonstrates that the production of a part in this work required 7 climb cutting actions, 8 conventional cutting actions. Each cutting action is separated by a brief air-cutting action, in which the machine pauses briefly between actions.

Dividing the time series

The milling machine G-code is used to divide the signal s into a series of shorter signals s^{ij}, corresponding to the signal produced during operation i with operation type j. The vibration signals contained 6142 points on average and the audio signals contained 49136 points on average.

Features are drawn from the Power Spectral Density (PSD) of the signal. Any change in the frequency content of the signal s^{ij} over time t^{ji} is assumed to be irrelevant to the prediction of tool condition. Welch's method (Welch, 1967) is used with a Hamming window to estimate the PSD for the signal. The time series signal is divided into successive blocks, with each block being defined by:

$$s_m^{ij}(n) = w(n)s(n + mR), \quad n = 0,1 \ldots M - 1, m = 0,1 \ldots K - 1$$

Where R is defined at the window hop size and K denotes the number of available frames. The hamming window is defined as follows:

$$h = hamming\ window$$

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

$$\widehat{P_{s_m}}^{ji}(f_k) = \frac{1}{M} \sum_{n=0}^{N-1} \left| \; s_m^{ij}(t_n)\, e^{-i2\pi n f \Delta t} \; \right|^2$$

The Welch estimate of the power spectral density is given by:

$$\hat{S}^{ji}(f_k) = \frac{1}{K} \sum_{m=0}^{K-1} \widehat{P_{s_m}}^{ji}(f_k)$$

The benefits of using Welches method over the Discrete Fourier transform are two-fold. The number of points in the resulting power spectrum is reduced and the random noise in the power spectrum is also reduced. A hamming window length of 512 points was chosen for the vibration data and a 2048-point window was chosen for the audio signal. The number of points in each power spectrum is equal to the number of points in the window. Therefore, the transformation from the time domain to the frequency domain represents a dimensional reduction of approximately 12 times in the acceleration signal and 24 times in the audio signal.

Figure showing

1) Time series
2) Labelled time series
3) PSD

Maybe Ranauk wants to comment about transmitting the PSD.

4 FEATURIZATION

The aim of this section is to identify a set of features, $\boldsymbol{x^{ji}} = \{x_1^{ji}, x_2^{ji} \ldots x_n^{ji}\}$ which are correlated with the condition of the milling machine tool. The first four features are derived from the Power Spectral Density (PSD) defined in Section 3. As the tool condition deteriorates the PSD changes. Several similarity measures are used to quantify the change in the PSD, including the Euclidian distance and the Fréchet distance. In each case the PSD, approximated by $\hat{s}^{ji}(f)$, is compared to a reference PSD, $\hat{s}^{j0}(f)$. The reference PSD represents the signal produced by a new tool. In this particular experiment the reference periodogram was obtained by averaging the first three periodograms for each action:

$$\hat{s}^{j0}(f) = \frac{1}{3}\left(\hat{s}^{j1}(f) + \hat{s}^{j2}(f) + \hat{s}^{j3}(f)\right)$$

3.1 Signal Power

The signal energy can be approximated by integrating the PSD with respect to frequency:

$$E_s = \int_{-\infty}^{\infty} |\hat{s}(f)|^2 \, df$$

Where E_s is the signal energy for the continuous signal $\hat{s}(f)$. In reality, the signal is expressed as a series of discrete points. Therefore, signal power energy can only be approximated from the Fourier coefficients:

$$E_s^{ji} \cong \sum_f \left|\hat{s}^{ji}(f)\right|^2$$

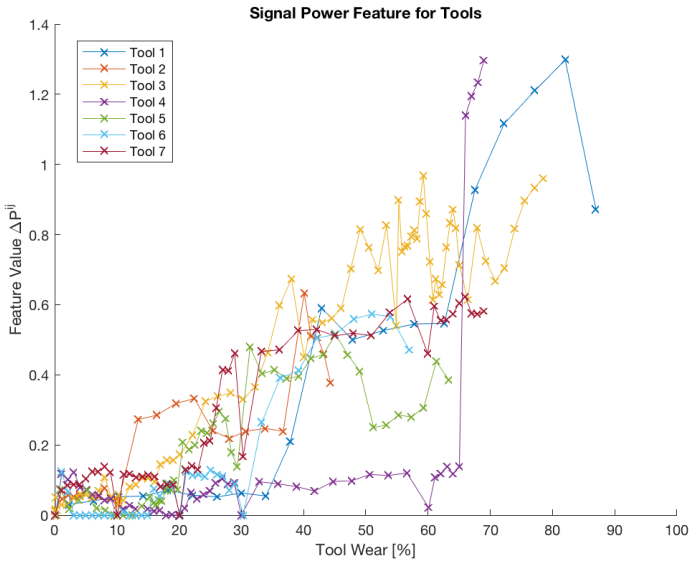
The signal power can be obtained by dividing the signal energy E_s^{ji} by the duration of the interval T_i.

$$P_s^{ji} = \frac{1}{T_i}\sum_f \left|\hat{s}^{ji}(f)\right|^2$$

The increase in signal power, between the reference signal $\hat{s}^{j0}(f)$ and signal $\hat{s}^{ji}(f)$ can be written as:

$$\Delta P^{ji} = P_s^{ji} - P_s^{j0} \qquad (2)$$

Where ΔP^{ji} is the normalized increase in signal power.



3.2 Magnitude of the transform function

In defining this feature we assume that the spectrogram $\hat{s}^{ji}(f)$ is related to the reference spectrogram $\hat{s}^{j0}(f)$ by some transform function, $a(f)$:

$$\hat{s}^{ji}(f) = a(f) \cdot \hat{s}^{j0}(f)$$

In discrete form we can write this as:

$$s_k^{ji} = a_k \cdot \hat{s}_k^{j0}$$

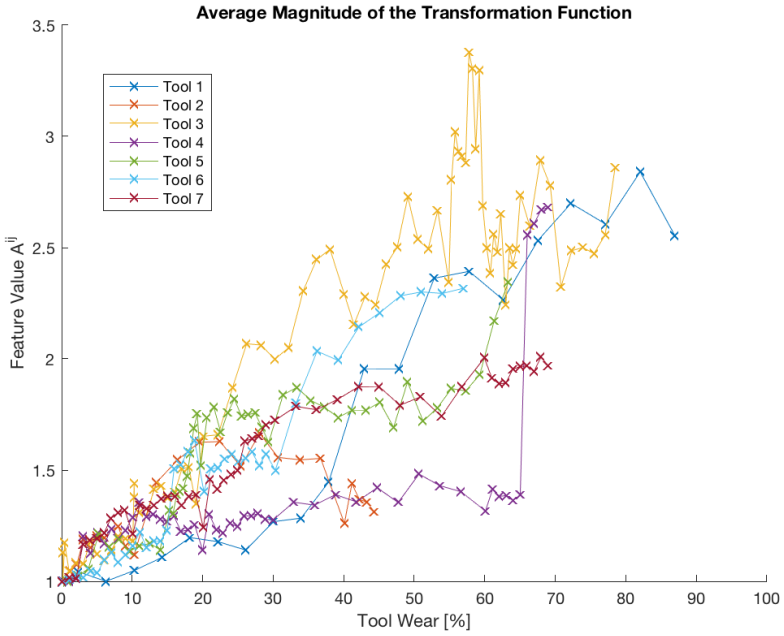
The value of the transformation function at frequency k is given by:

$$a_k = \frac{s_k^{ji}}{\hat{s}_k^{j0}}$$

Summing the transformation function over the frequency domain:

$$\Delta A^{ji} = \sum_{k=1}^{n} \frac{s_k^{ji}}{\hat{s}_k^{j0}}$$

Where A^{ji} is a feature representing the magnitude of the transform function.



3.3 Fréchet distance between spectograms

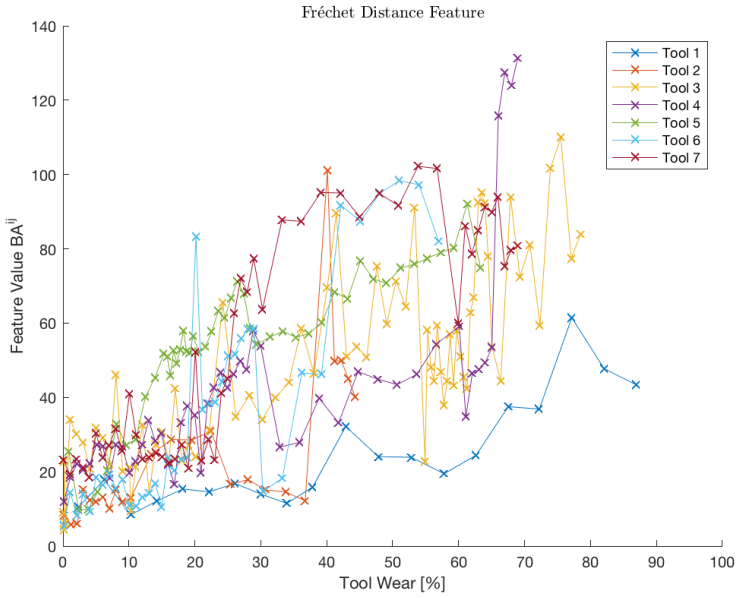
In defining this feature we treat the spectrogram as a curve in two-dimensional Euclidian space, with frequency as the first dimension and amplitude as the second. We define a new feature D^{ij}, as the Fréchet distance between the spectrogram $\hat{s}^{ji}(f)$ and the reference spectrogram$\hat{s}^{ji}(f)$. The Fréchet distance is a measure of the similarity of two curves that accounts for both the location and ordering of points along the curve. An intuitive definition of the Fréchet distance is the minimum length of the leash required to connect a dog and its owner as they walk along two curves without backtracking.

The formal definition of the Fréchet distance between the two continuous curves P and Q is as follows. Let ϕ_n be the set of all continuous non-decreasing functions ϕ from [0,1] onto $[0, n]$. The continuous Fréchet distance between curves P and Q with $|P| = n$ and $|Q| = m$ is given by:

$$\delta_F(P, Q) = \inf_{\substack{\phi_1 \in \phi_n \\ \phi_2 \in \phi_m}} \max_{t \in [0,1]} \left\| P\left(\phi_1(t)\right) - Q(\phi_2(t)) \right\|$$

$\phi_n$where $\|\cdot\|$ denotes the Euclidian distance. We define a new feature B^{ij} as the discrete Fréchet distance between the two curves:

$$B^{ji} = \delta_F(\hat{s}^{ji}, \hat{s}^{j0})$$



3.4 Kurtosis coefficient

Several authors have indicated that the condition of the tool is related to the kurtosis coefficient of the acoustic signal (ref). The kurtosis coefficient provides a way to identify sudden changes in the time series signal (ref). The kurtosis coefficient is calculated on the device at regular intervals *i:*

$$\kappa^{ji} = \frac{E[(s^{ji}(t) - \mu_s)^4]}{E[(s^{ji}(t) - \mu_s)^2]^2}$$

Where the signal mean μ_s and is defined as:

$$\mu_s = E[\, s^{ji}(t)]$$

The kurtosis coefficient is also calculated for the reference time-series signal, s^{j0}:

$$\kappa^{j0} = \frac{E[(s^{j0}(t) - \mu_s^0)^4]}{E[(s^{j0}(t) - \mu_s^0)^2]^2}$$

A new feature, K^{ij} is defined as the ratio of the kurtosis coefficient of signal s^{ji}, normalized by the kurtosis coefficient of the the reference signal κ^{j0}.

$$K^{ij} = \frac{\kappa^{ji}}{\kappa^{j0}}$$

4. DEVELOPING A GAUSIAN PROCESS MODEL TO PREDICT TOOL CONDITION

Guassian process regression (GPR) is supervised machine-learning method that performs particularly well with noisy data. GPR uses a Gaussian process (GP) as a prior to describe the distribution on the target function $f(\boldsymbol{x})$. A GP is defined as a collection of random variables, any finite set of which has a joint Gaussian distribution [1] (rephrase according to original ref, not Jinkyoo et al). The GP can be fully specified by its mean function $m(\cdot)$ and covariance function $k(\cdot,\cdot)$, since it is just a multivariate Gaussian distribution over function $f(\cdot)$. In a GP, $m(\cdot)$ and $k(\cdot,\cdot)$ are not constant parameters but functions incorporating prior knowledge about the unknown function $f(x)$ (ref MIT Press).

$$p(f^{1:n}) = GP\big(m(\cdot), k(\cdot,\cdot)\big) \tag{1}$$

The mean function $m(\cdot)$ captures the overall trend in the target function value, and $k(\cdot,\cdot)$ is used to approximate the covariance by representing the similarity between the data points. In GPR, $m(\cdot)$ is often chosen to be a zero function (ref). In this case the target function is fully described by the covariance kernel function $k(\cdot,\cdot)$. The type of kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ used to build a GPR model can strongly impact the accuracy of the prediction model. In this study the Automatic Relevance Determination (ARD) squared exponential covariance function was chosen:

$$k(\boldsymbol{x}^i, \boldsymbol{x}^j) = \gamma \exp\left(-\frac{1}{2}(\boldsymbol{x}^i - \boldsymbol{x}^j)^T \mathrm{diag}(\boldsymbol{\lambda})^{-2}(\boldsymbol{x}^i - \boldsymbol{x}^j)\right) \tag{2}$$

where the kernel function is described by the hyper-parameters, γ and $\boldsymbol{\lambda}$. The signal variance hyper-parameter γ quantifies the overall magnitude of the covariance value. The hyper-parameter vector $\boldsymbol{\lambda} = (\lambda_1, \dots \lambda_i, \dots, \lambda_m)$ is quantify the relevancy of the input features in $\boldsymbol{x} = (x_1, \dots x_i, \dots, x_m)$ when predicting the response y.

EXPLAIN THE NOISE MODEL HERE

4.1 Training Procedure

The hyper-parameters $\boldsymbol{\theta} = (\sigma_\epsilon, \gamma, \boldsymbol{\lambda})$ are chosen to maximize the marginal likelihood of observations in a given training data set training data $\boldsymbol{D}^n = \{(\boldsymbol{x}^i, y^i) | i = 1, \dots, n\}$:

$$\begin{aligned} \boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta}}{\mathrm{argmax}} \log p\,(\boldsymbol{y}^{1:n}; \boldsymbol{\theta}\,) \\ &= \underset{\boldsymbol{\theta}}{\mathrm{argmax}} \left(-\frac{1}{2}(\boldsymbol{y}^{1:n})^T(\mathbf{K} + \sigma_\epsilon^2\mathbf{I})^{-1}\boldsymbol{y}^{1:n} - \frac{1}{2}\log|\mathbf{K} + \sigma_\epsilon^2\mathbf{I}| - \frac{n}{2}\log 2\pi\right) \end{aligned} \tag{3}$$

where $\mathbf{K}$ is the covariance kernel matrix defined as $\mathbf{K}_{ij} = k(\boldsymbol{x}^i, \boldsymbol{x}^j)$. After optimizing the hyper-parameters the length-scale hyper-parameter vector $\boldsymbol{\lambda}$ can be used to determine the relevance of each feature, x_i . A small value of λ_i indicates high relevancy of x_i while a large value indicates low relevancy.

4.2 Scoring Procedure

After choosing the kernel type and optimizing the hyper-parameters, the GPR model is referred to as 'trained'. The model can be used to predict the hidden function value $f(\boldsymbol{x}^{new})$ based on a new observation $\boldsymbol{x}^{new} = (x_1 \dots x_i \dots x_m)$. The observed outputs $\boldsymbol{y}^{1:n} = \{y^1, \dots, y^n\}$ and the hidden function value follow a multivariate Gaussian distribution.

$$\begin{bmatrix} \boldsymbol{y}^{1:n} \\ f^{new} \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} (\mathbf{K} + \sigma_\epsilon^2\mathbf{I}) & \boldsymbol{k} \\ \boldsymbol{k}^T & k(\boldsymbol{x}^{new}, \boldsymbol{x}^{new}) \end{bmatrix}\right) \tag{4}$$

Where the hidden function value $f(\boldsymbol{x}^{new})$ as is denoted as f^{new} for brevity.
EXPLAIN HOW $\mu(\boldsymbol{x}^{new}|\boldsymbol{D}^n)$ and $\sigma^2(\boldsymbol{x}^{new}|\boldsymbol{D}^n)$ can fully describe the posterior distribution on the response f^{new}.

EXPLAIN THE GPR PROCESS IN TERMS OF TOOL CONDITION, OR ADD SOME EXPLAINATION HERE.

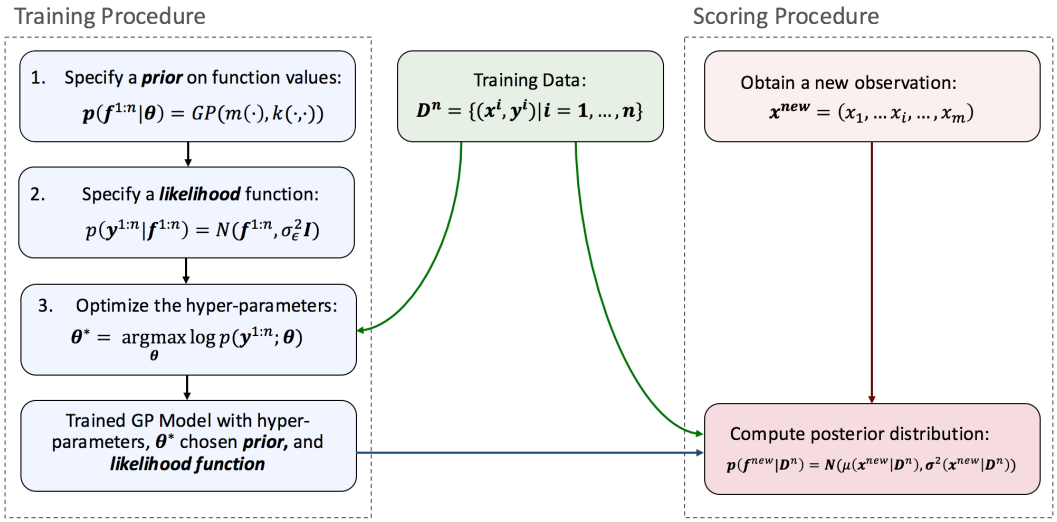


Figure 2. Flowchart showing GPR training and scoring procedure

5. RESULTS

It was hypothesized that the condition of the tool could be predicted from the acoustic and acceleration signal produced by the milling machine. the frequency content of the acceleration signal (ref), as well as the spectral energy density of the both the vibration and acceleration signals. This section describes the relevance of each feature to the tool condition model, as well as the overall performance of the model. We also need to talk about the results from block-wise analysis. For example, what actions resulted in better condition prediction.

5.1 Evolution of spectral energy

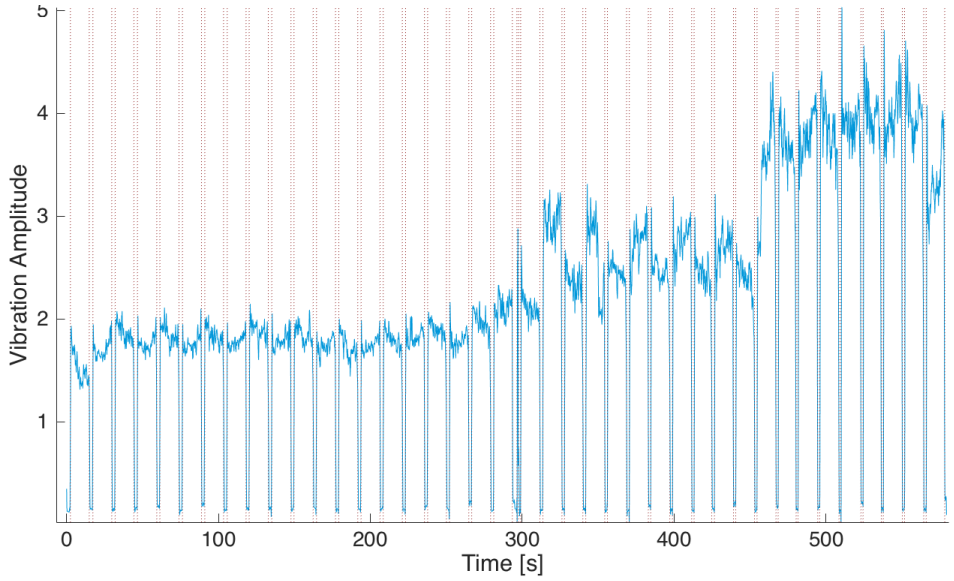


Figure 3. To be replaced with a similar plot showing spectral energy against time

Figure x indicates how the spectral energy increase as the tool condition decreases. The green line indicates how the condition of the tool was labelled after being manually examined by a microscope

5.2 Evolution of the frequency content

As the condition of the milling machine deteriorates the audible sound produced by the milling machine changes. This change in frequency content can be detected by an experienced operator, and corresponds to a change in both signal energy and frequency content.

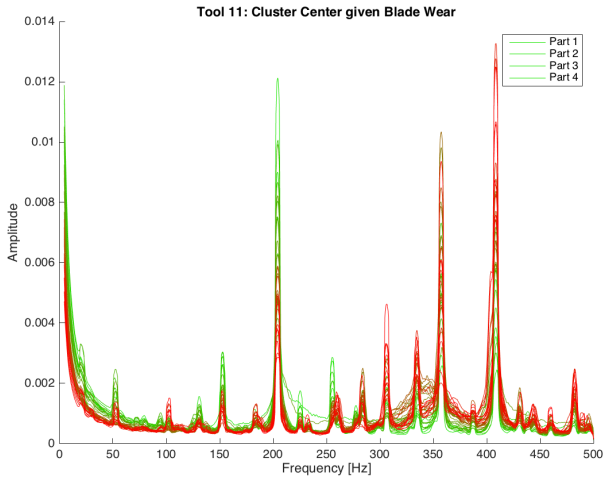


Figure 4. Frequency content of the vibration signal for the first part (sharp blade) and subsequent blades. The color indicates the condition of the blade as determined by inspecting the blade with a microscope. Note that the signal energy has been normalized to 1, to ensure that the graph only reflects change frequency distribution. My plan is to subtract the baseline signal from this figure so that it only shows the change. I think it will look great.

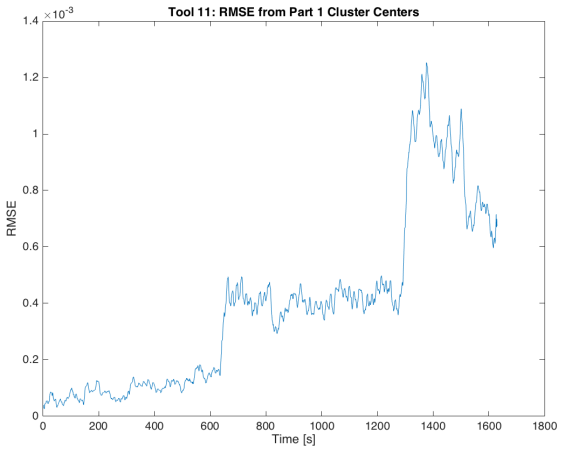


Figure 5. RMSE change in the frequency content over time. The frequency content produced whilst producing the first part is compared to the frequency content whilst producing subsequent parts

Figure 6 demonstrates that the frequency content of the vibration signal changes as the tool condition deteriorates. In particular, the amplitude of the 200 Hz mode decreases relative to the spectral energy. The 200 Hz frequency value is of particular interest, because it represents the frequency at which each flute makes contact with the steel block.

Figure 7 quantifies the transient nature of the frequency content, and demonstrates that the vibration frequency content changes as the tool deteriorates. The figure also indicates that the vibration frequency content changes abruptly at two points during the life of the tool. These two points are likely to correspond to sudden damage to one of the blade flutes, but this was never confirmed.

5.2 Tool condition prediction

The results shown in this section are obtained from data set 1-15. I want to repeat the experiments again using the setup we used with Dave Colleran. It seems that the more recent results are far more consistent and predictable (especially now that we are using coolant). Once we start using the new data, the length of the tool life will be more uniform. **Then I can plot tool condition (100%==good, 0%==bad) against time. I will plot time on the x, and human labels/GP predictions on the y. Also can plot GP standard deviation which will look incredible!**

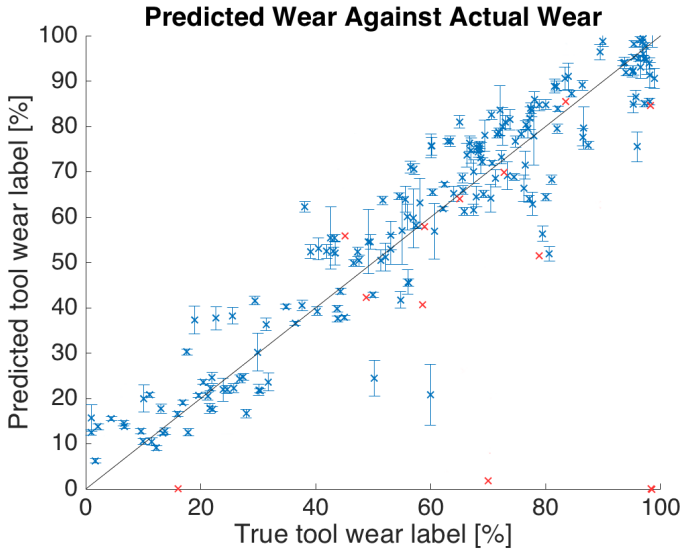


Figure 6. True tool wear as manually labelled using microscopic images, plotted against the predicted tool wear using Gaussian Process Regression (for the entire test data set)

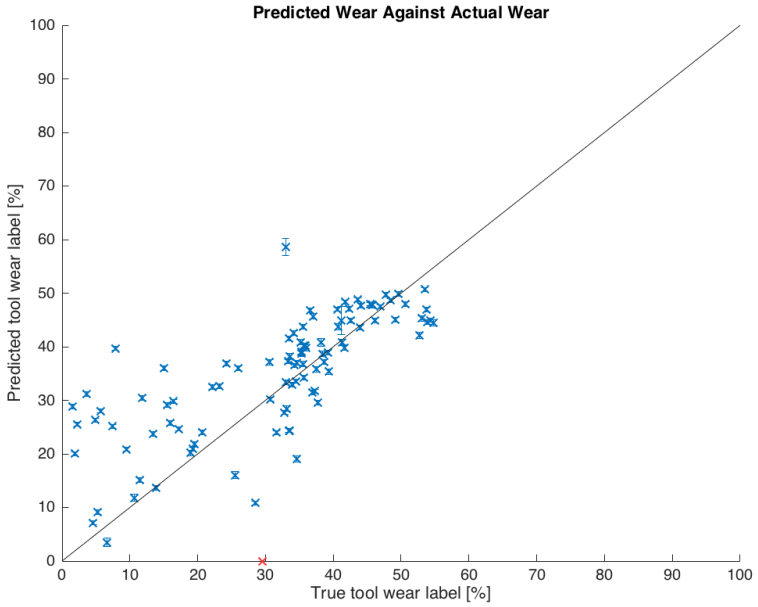


Figure 7. True tool wear as manually labelled using microscopic images, plotted against the predicted tool wear using Gaussian Process Regression (for tool 8 data set)

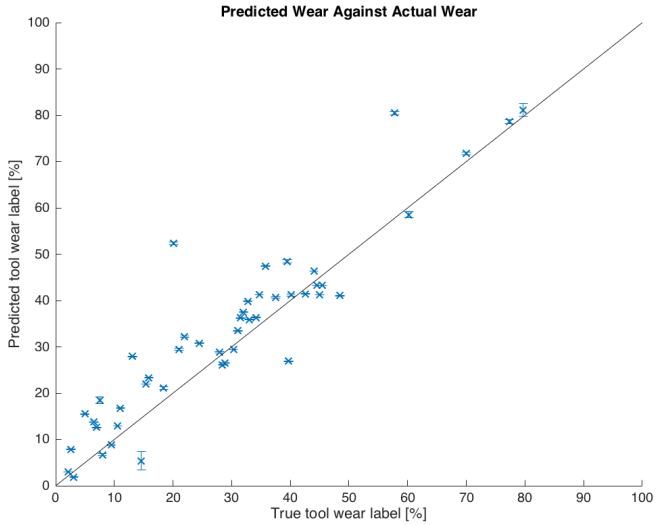


Figure 8. True tool wear as manually labelled using microscopic images, plotted against the predicted tool wear using Gaussian Process Regression (for tool 12 data set)

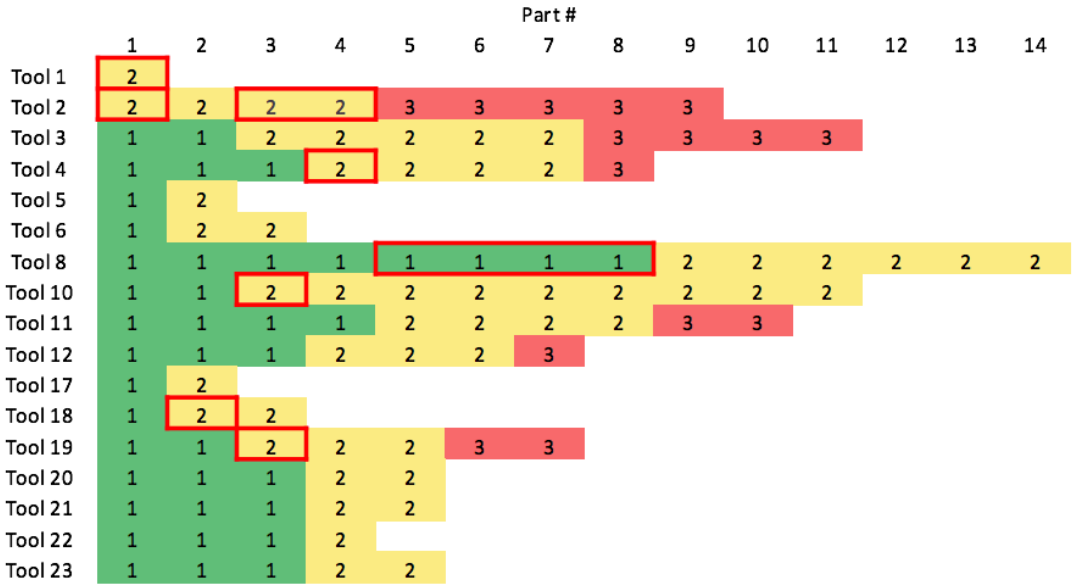


Figure 9. Tool condition predictions (1=good, 3=worn) as predicted by the GP model. Each square represents a single part. Squares with red a red outline indicate where the GP model predicted a different label than the label assigned manually assigned from the microscope photo.

6. DISCUSSION AND CONCLUSIONS

7. REFERENCES

1. C. Rasmussen, and C. Williams, "Gaussian Process for machine learning", MIT Press, 2006.
2. R. M., Neal. (1996). *Bayesian learning for neural networks*. Springer-Verlag, New York.
3. R. Bhinge, J. Park, N. Biswas, M. Helu, D. Dornfeld, K. Law and S. Rachuri, "An Intelligent Machine Monitoring System Using Gaussian Process Regression for Energy Prediction," *in Proceeding of IEEE International Conference on Big Data* (IEEE BigData 2014), 2014, Washington, DC.
4. J. Park, R. Bhinge, N. Biswas, M. Srinivasan, M. Helu, S. Rachuri, D. Dornfeld and K. Law, "A generalized data-driven energy prediction model with uncertainty for a milling machine tool using Gaussian Process," *in Proceeding of ASME 2015 International Manufacturing Science and Engineering* Conference, 2015, Charlotte, NC.
5. Teramura, K. Okuma H. Taniguchi Y, Makimoto, S. Maeda S. "Gaussian Process Regression for Rendering Music Performance", *in Proceeding of the 10th International Conference on Music Perception and Cognition,* 2015 Sapporo, Japan.
6. V. Tresp, "A Bayesian committee machine," *Neural Computation,* 12(11), 2000, pp. 2719–2741.
7. D. Nguyen-tuong, and J. Peters, "Local Gaussian process regression for real time online model learning and control," *Advances in Neural Information Processing Systems*, MIT Press, 2008.
8. J. Shi, R. Murray-Smith, and D. Titterington, "Hierarchical Gaussian process mixtures for regression," *Statistics and Computing*, 15, 2005, pp. 31-41.
9. A. Ranganathan, M. Yang, and J. Ho, "Online sparse Gaussian Process regression and its applications," *IEEE Transactions on Image Processing*, 20(2), 2011, pp. 391-404
10. H. Xiao, and C. Echert, "Lazy Gaussian Process Committee for Real-Time online regression," *in Proceeding of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-13)*, Bellevue, WA, USA, 2013.
11. J. Quiñonero-Candela and C. Rasmussen, "A unifying view of sparse approximate Gaussian Process regression," Journal of Machine Learning Reseach, 6, 2005, pp. 1939-1959.