Neural Network Based Chatter Detection and Control on CNC Milling Machine

Kaan Bahtiyar[1], Mert Ozdemir[2] and Satyam Bhawsinghka[3]

***Abstract*— Chatter is undesirable self-excited vibrations which occurs during machining processes and leads in damage to the workpiece and the machine elements. Therefore, having a stable cutting is one of the most important features for all manufacturing processes. Chatter can be prevented either by adjusting parameters to achieve stable cutting on detection or by setting them appropriately before the process starts. Since setting stable cutting parameters before operation requires pre-processing experiments and simulations, online chatter detection is preferred. Although existing methods can detect chatter in real-time, they have their own disadvantages in terms of resolution, tuning parameters, late detection. In this paper, a neural network has been trained without the use of machine tuning parameters on time sampled normalized acceleration data of the machine tool to detect chatter in real-time by predicting in advance the onset of chatter at the end of a time window, upon which the spindle speed is changed to stabilize machining conditions to prevent chatter. Experimental results are presented and compared with the state-of-art chatter detection algorithms. The results confirm accuracy and robustness of the proposed method for detection of chatter.**

I. Introduction

Most machining processes are subject to undesired vibrations in the form of forced and chatter vibrations that affect the quality of the process. Although forced vibrations are a result of cutting forces, chatter is self-excited form of vibration which grows exponentially and becomes destructive once triggered. Chatter has detrimental impact in the industry as it becomes a limiting factor and forces the operation to be performed at cutting speeds below the machine's capabilities. Also, these vibrations not only result in poor surface finish but may also lead to tool breakage [1]. Therefore, setting correct machining parameters before operation, or stopping the process rapidly and adapting the parameters to stable pairs when chatter occurs is critical for machining applications.

One way to detect chatter boundary can be performed by modeling system dynamics of machining process. This can be done by using time domain simulations. However, the border between stable and unstable cutting can not be distinguished from these simulations clearly since the system involves some non-linear properties which can not be modeled easily such as friction between tool and workpiece and thermo-mechanical effects on chip formation [2].

*Course: Learning Based Control
*Instructor: Dr. Kagan Tumer

[1]`bahtiyak@oregonstate.edu`
[2]`ozdemirm@oregonstate.edu`
[3]`bhawsins@oregonstate.edu`

Setting machining parameters correctly to prevent unstable cutting before the operation is another method to avoid chatter, and attracts lots of scholars on this topic [3]. This is performed by obtaining stability lobe diagram, and setting the correct pair of spindle speed and axial depth of cut which are the parameters that affect chatter stability [4]. An example for stability lobe diagram is given in Figure 1. The graphical representation of a stability lobe diagram shows the limit depth of cut to be set before process for each spindle speed to have stable cutting. However, in order to predict the diagram analytically, some experiments need to be performed on the machine for cutting coefficients identification, and frequency response (FRF) identification which are used in calculations to generate a stability lobe diagram [5]. These experiments take time and are required to be repeated for each tool-workpiece pair during which the machine wouldn't be operational. Also, the quality of identified FRF is depend on experiment which should be designed carefully and should be conducted without interrupting system dynamics. Hence, experiments should be repeated couple of times for each tool-workpiece pair to check the correctness of the result of the experiment, where one of those experiment and analyzing their outputs takes lots of time [6]. Wasting time for each tool-workpiece pair is terrible for a high-volume manufacturing company since the aim of the CNC workshops is to use machines 24 hours a day. The cost of this situation can be calculated in terms of time. Therefore, performing such experiments result in decreased productivity and efficiency in the industry.

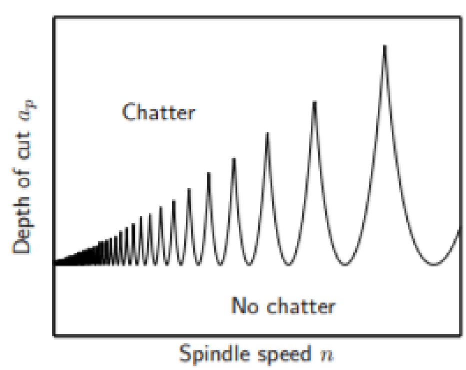


Fig. 1. Stability Lobes Diagram

Instead of performing those experiments and pre-processing, chatter is detected on-line in most cases. There are some real-time methods [7] which detect chatter based on time and frequency domain approaches. Those algorithms aim to detect chatter right before chatter marks are left on the surface of the workpiece. However, frequency domain approaches are based on Fourier Transform (FT) which requires a batch of data to analyze at once. Also, frequency domain approaches detect chatter vibrations only after the vibration has fully grown. This is due to the fact that FT is not real-time and requires a batch of data to analyze power spectrum of the signals. Moreover, resolution of FT is based on the size of window that data buffered and it requires larger batch sizes, i.e., larger delays in detection. On the other hand, time domain analysis for chatter detection requires lots of tuning parameters which causes the detection to be machine/cutting condition specific. The effect of tuning parameters affects quality of detection which may give false alarms or may miss chatter due to wrong tuning.

The contributions of this paper are to
(1) remove the effect of tuning parameters in algorithms used to detect chatter,
(2) provide chatter detection using only acceleration data, regardless of dynamic behavior of tool-workpiece pair,
(3) eliminate resolution issue.

In this paper, a neural network (NN) is deployed to detect the early onset of chattering by taking acceleration data from time domain milling simulation. The NN is trained on acceleration values at discrete intervals in a time window. The acceleration values are normalized by a value that enables the NN to work for different machining conditions and machines. A concept of time gap has been introduced that enables the prediction of early onset of chatter. The implemented network enables to have generalized chatter detection method since it only use acceleration data in a windowed fashion. Once the chatter is detected, spindle speed variation technique is used to suppress chatter vibrations.

II. Background and Related Work

A. Chatter Vibrations

Milling process suffers from both forced and chatter vibrations concurrently where forced vibrations are a natural result of tooth entry/exit to the workpiece but chatter vibrations are self-excited and harmful for safety and quality of the machining processes [4]. Figure 2 shows the difference stable and unstable surface finish where how bad surface quality becomes in chattering condition can be seen.

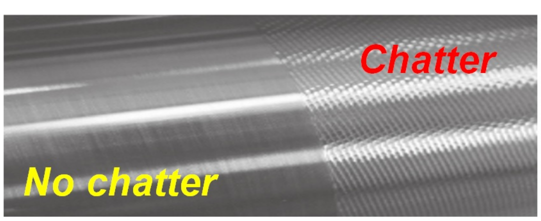


Fig. 2. Detail of a workpiece without and with chatter marks

Forced vibration component is because of the nature of immersion of milling tool cutting edges into the workpiece. Therefore, forced vibrations are related to angular rotation frequency of the cutter, and they have a wide spectrum due to the effects of structural dynamics of the tool, shape of cutting edge [4]. The frequency spectrum of the cutting process is governed mainly via tooth passing frequency, i.e., spindle speed. The resultant cutting force profile can be reconstructed by summing the whole spectrum. In practice, the spectrum contains only several harmonics of the spindle speed, and the governed cutting force is obtained by summation of those. One of the resonances of tool/workpiece system is triggered by cutting forces, and chatter occurs close to the resonant frequency of the structure. The reason of the chatter phenomena is that a wavy surface finish left from previous tooth is removed during the current tooth which also leaves a wavy surface due to the structural vibrations [8]. The phase shift between previous and current waves leads the chip thickness to grow exponentially while oscillating at the chatter frequency [5]. The frequency of the chatter may have a fundamental frequency and several side bands as a result of structural dynamics of the tool [7]. Figure 3)a) shows a frequency spectrum of a machining process without chatter, 3)b) shows with the onset of the chatter, and 3)c) presents with grown chatter case. As can be seen, in stable cut, all of the energy is at spindle rotation frequency and its harmonics while the energy begins to be dominated by chatter frequency and its side bands for unstable case. It should be noted that, chatter vibrations grows exponentially and dominates the whole process instantaneously.

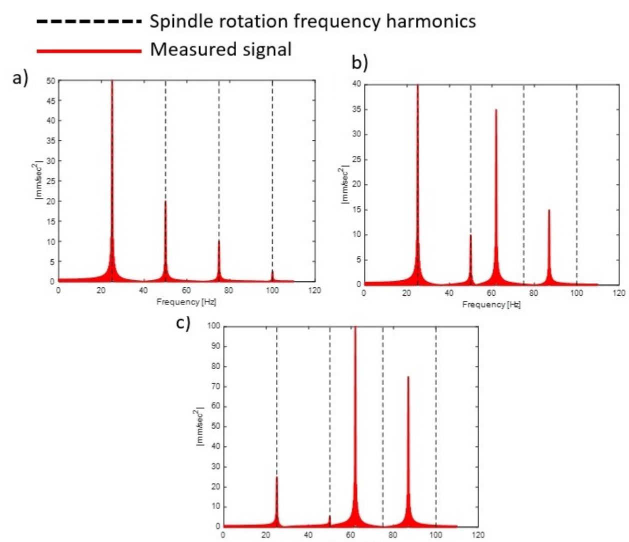


Fig. 3. Frequency spectrum for a) stable cutting, b) chatter onset, c) grown chatter

B. Chatter Detection

Chatter detection is performed real-time and relies on the feedback from the sensors such as accelerometer, microphone, force sensor etc, and after the chatter is detected, controller acts on spindle speed until the feedback becomes stable. There are lots of scholars focus on chatter detection and control. However, those methods have some drawbacks as well. Although chatter detection and control are dependant parts to each other, chatter detection is the main concept governed in this project.

As shown in Figure 3, chatter onset can be detected by using frequency spectrum i.e. the moment when the process begins to be dominated by chatter frequencies. Currently, frequency domain approaches are used in the literature where Fourier Transform of the measured vibration or sound signal is obtained and analyzed. Forced vibration related energy is calculated from frequency spectrum which is performed by summing the energies at the spindle rotation frequency and its harmonics. Since the chatter shows up between spindle harmonics, the rest of the whole spectrum than spindle frequency harmonics is the energy consumed by chatter and noise. Therefore, by looking at the ratio between chatter energy over total energy, detection of the chatter is decided [9]. However, there is no strict ratio for detection of the chatter, but generally, 50% is seen suitable for most applications which means half of the total energy is consumed by chatter in the process although the threshold should be set by looking at the surface finish.

Even though this has become the standard procedure [10], it can detect chatter only after it occurs since Fourier analysis requires a windowed data to accurately analyze the signal. Moreover, keeping resolution of Fourier analysis high is another issue for this method which requires the window size is large enough to capture whole frequencies which brings a delay to the detection process. Besides, both time and/or frequency domain approaches requires to tune their parameters, and the number of the parameters are really much to make the algorithms not to be machine/process specific.

In the [7], author extracts forced vibration components by using Kalman Filter and uses energy ratio approach to determine if the process is chattering or not. Although the method exhibits promising results, Kalman Filter is required to be tuned before application and if the tuning is not done proper, the filter tracks the measured signal instead of forced vibration part. Besides, the indicator is biased since the power of chatter frequency is used. Likewise, in [11], they use autocorrelation coefficient as an identifier for chatter. This method gives proper results when the chatter frequency is clearly sought in-between spindle rotation frequency harmonics. However, if the chatter frequency occurs very close to spindle rotation frequency harmonics, the algorithm may fail since for a sliding windowed data, autocorrelation can not capture small frequency slippages, and gives high/stable results although the process is unstable.

Unlike what has been mentioned so far, [12] uses parametric modeling for chatter detection which basically fits a transfer function to the measured data to extract chatter frequencies. Once the model is fitted, chatter detection is performed on constructing the model reflects only chatter part. However, this method may fail for sound signals which may be unsteady at the onset of the chatter. On the other hand, [3] uses notch filter to extract the chatter component which is not suitable since it brings delay to detection and needs to be tuned from application to application as well. Moreover, the subspace-based estimation used in [3] is not suitable for real-time since it has high computational cost although it is accurate and robust.

The method [13] seems pretty promising and independent of machining conditions. It uses moving variance operator to calculate the sum of the whole power spectrum of the measured vibration, and calculates moving FT to calculate the power of forced vibration part. However, once the signal window size is decreased, there is a fluctuation on indicator due to FT and variance operators.

C. *Chatter Detection by Learning Based Methods*

[14] uses hybrid machine learning based chatter detection model where machine learning is used to support chatter detection algorithm with classifying cutting stage probability to increase efficiency of the chatter detection method in tool entrance/exit. However, the method requires much computational capacity to perform real-time and generalized for different cutting conditions. Also, it still searches peaks on frequency spectrum which has a leakage issue decreasing the resolution and accuracy of the method.

[15] proposed a method to detect chatter by utilizing neural network. The method is composed of 5 different stage; (1) data acquisition, (2) signal processing, (3) features generation, (4) features selection, classification(5). The data is collected by implementing three accelerometers on the machine with three different axis. Although the stability of the machining operation is highly depends on the structural dynamics of the machine, it is observed that chatter occurs at close to the natural frequencies. Therefore, the data corresponding close to the natural frequencies are filtered. The metrics that are influenced by chatter phenomenon and related with milling states are generated, then they are selected according to their significance on the chatter. The metric selection is done by using relative entropy measure. They are ranked from best to worst with this measurement technique. The authors utilize two neural network approaches, Multilayer Perceptron with two hidden layers and Radial Basis Function to classify the chatter occurs whether or not. Supervised learning is used for this particular application, which means for every input there exist a desired output. The major drawback of this method is that it requires FRF of the machine tool-workpiece pair to detect chatter because they use only the data close to the natural frequency. However, the FRF identification experiments need to be done to obtain natural frequencies. Besides, filtering in real-time processes brings a delay to the data which may cause to miss the chatter onset. Moreover, feature generation and selection parameters may need to be tuned for different applications.

In this paper, a strategy to detect chatter by using only acceleration data is proposed. It is promoted that chatter detection can be performed at its early onset stage by using NN which is trained over time domain milling simulation. Besides, by normalization of the windowed time series data, the network is aimed to be machine/condition independent. Also, the only parameter window size is provided to be smaller with respect to frequency domain approaches since NN works in time domain.

III. Methodology

A. Simulation

In this section, a time domain milling simulation is detailed for the purpose of using in training data generation. Since chatter is destructive phenomena, having a database on chattering conditions and training a neural network on it is not viable because collecting that database may ruin the workpieces, tools, and the machine equipment. Therefore, a simplified simulation is conducted at this point.

Time domain milling simulation is a demonstration of real-time milling process. It takes machining conditions (spindle speed, depth of cut), machine parameters (frequency response, model parameters of the machine), desired time step and simulation time as input whereas it generates cutting forces and acceleration data of the tool with respect to time. In this simplified simulation, the inputs are set before the simulation and can not be changed during operation. Since the process mechanics is the same with all milling machines, the simulation is valid for all type of milling machines and CNCs as long as their specific parameters are provided as input. However, the data generated for one machine parameters can be used to train a neural network which is used to detect chatter for all milling machines because the generated acceleration data is sum of sinusoidal signals (See Figure 3) and the shape of chattering acceleration data resembles each other for all machines, only their fundamental spindle and chatter frequencies change. Therefore, even the simulation is obtained for one machine, the acceleration data can be used as generalized training data.

In this report, the simulation is created on MATLAB and the steps can be given as follows [1].

- The current chip thickness is calculated by using the vibration of the current and previous teeth at the selected tooth angle.
- The cutting force is calculated.
- The new displacements are obtained by using the cutting force.
- The tooth angle is incremented and the process is repeated.

In the simulation, first of all, the instantaneous chip thickness is calculated by using nominal tooth angle-dependent chip thickness, the current normal direction of vibration, and the vibration of the previous tooth. The formula for chip thickness ($h(t)$) calculation can be given as follows.

$$h(t) = f_{\mathrm{t}} sin(\phi) + n(t-\tau) - n(t) \tag{1}$$

where τ is the tooth period, f_{t} is the feed rate, $f_{\mathrm{t}} sin(\phi)$ is nominal chip thickness, $n(t)$ is the current and $n(t-\tau)$ is the previous tooth's vibration. Once the instantaneous chip thickness ($h(t)$) is obtained, the tangential and normal components of the instantaneous resultant force are calculated as following.

$$F_{\mathrm{t}} = K_{\mathrm{t}} bh \tag{2}$$

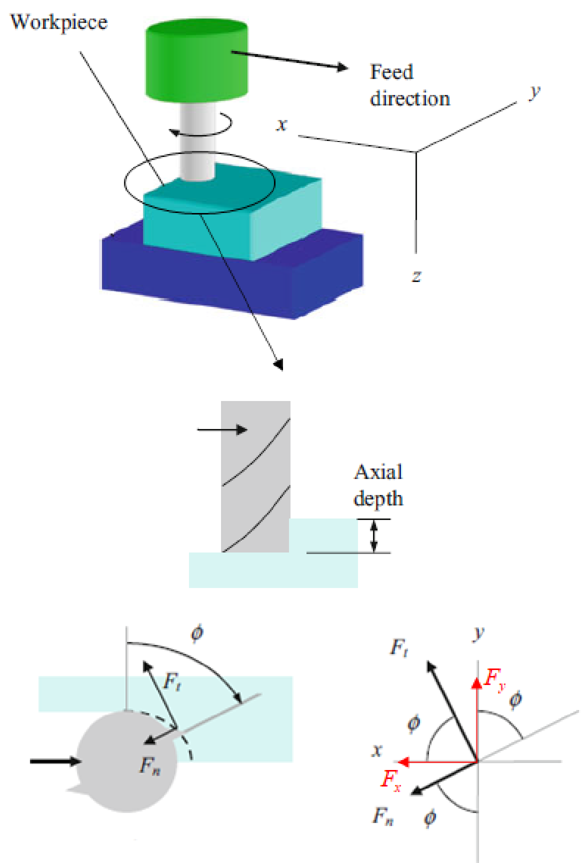


Fig. 4. Milling force illustration [1]

$$F_{\mathrm{n}} = K_{\mathrm{n}} bh \tag{3}$$

where n denotes normal, t denotes tangential component, b is the axial depth of cut, h is the current chip thickness, K_t and K_n are cutting coefficients. The illustration for the cutting forces can be seen in Figure 4. Here, it should be noted that, if the current tool vibration in the normal direction is larger than the surface location (which is sum of nominal chip thickness and the vibration of the previous tooth at the same angle), the chip thickness becomes negative which means the edge of cutter is out of the workpiece. Therefore, the force components are set to zero since that indicates there is not cutting. Also, it should be noticed that those tangential and normal forces are based on tool cutter geometry angles, so they should be projected onto global x and y directions of the machine. The tangential and normal force components are used to calculate resultant force, and projected forces onto x and y directions of the global coordinate system of machine. Then, displacements, velocities, and accelerations are calculated for x and y directions by solving following differential equations with numerical methods.

$$m_{\mathrm{x}}\ddot{x} + c_{\mathrm{x}}\dot{x} + k_{\mathrm{x}}x = F_{\mathrm{x}} \tag{4}$$

$$m_{\mathrm{y}}\ddot{y} + c_{\mathrm{y}}\dot{y} + k_{\mathrm{y}}y = F_{\mathrm{y}} \tag{5}$$

As can be interpreted that solution of the differential equations generates sinusoidal signals which is used for chatter detection as explained above. Those sinusoidal signals are distinguished as stable and unstable by looking at the frequency of those components, and if the dominant frequencies are in-between spindle rotation frequency harmonics, it is called chatter.

In this paper, simplified time domain simulation is used to generate the training data for neural network. Simulation data is created at a sampling rate of *1 kHz*. In order to obtain various results, randomized simulations are performed in a way that has random spindle speed, random length of simulation, and random start time of chatter. Also, chattering is randomized for the simulation. It is ensured that there is enough time for chatter to grow if there is going to be chatter. The constraint for randomizing are set for the spindle speed between *10-150 Hz*, whereas it is *6-10 seconds* for the length of simulation, and the start time of chatter is randomized after *5th sec*. Thus, *3000* random machining conditions are generated and the results are saved so that neural network is fed with those time-domain acceleration signals, and the start time index of chatter. Besides, since having chatter is also randomized, the network is also trained with stable data to eliminate bias toward to chatter.

B. Data Preparation

The acceleration data from the simulation is generated as a time series data. The simulation generates the acceleration values at every *1 millisecond (ms)*, i.e. *1 kHz*, and those values are saved to an Excel file in tabular form. Along with the acceleration signal, chatter occurrence time ($t_{chatter}$) is also used to prepare data for the NN model.

The input data for the NN is prepared by taking absolute acceleration values at regular time intervals for a time window of *0.2 seconds* (t_{window}). For the sake of comparison, the acceleration values are down sampled for two different time intervals, *5* and *10 ms*, i.e. *200* and *100 Hz*. Therefore, for the windowed data (t_{window} is *0.2 seconds*), only *40* and *20* samples are considered instead of having computation of all *200* samples. This decreases the computational complexity of the network. Also, there is a concept of time gap (t_{gap}) introduced in this approach which makes this model predictive i.e. the ability to detect the early onset of chatter. The extent of t_{gap} governs how well in advance the model will be able predict the onset of chatter. For the sake of comparison, three different models have been trained for t_{gap} as *0*, *0.2* and *0.4 seconds*.

There are two classes of acceleration data, one which indicates the presence of chatter and one that does not. The label for the prior will be *1* and the latter will be *0* in the model for NN. To create the feature set from the simulation output for data labelled *1*, the acceleration values are taken for the time window starting at time:

$$t = t_{\text{chatter}} - t_{\text{window}} - t_{\text{gap}} \tag{6}$$

and ending at time:

$$t = t_{\text{chatter}} - t_{\text{gap}} \tag{7}$$

sampled at the specified time interval. For data labelled 0, the acceleration values are extracted from t = *10 ms* to *10* + t_{window} *ms* at specified time intervals. It is assumed that there is no chattering at the start of the simulation and this data will indicate that there is no chattering. This assumption is valid because in the simulation the chattering was induced at a later stage. The time window begins at *10th ms* and not at *0* in order for signal to stabilize. The acceleration values extracted from various simulations as described above are the feature set and they along with the labels indicating chattering become the dataset on which the models are trained.

C. Training the Neural Network

The feature set extracted from the simulation as described above is normalized using the *StandardScaler* function of *sklearn.preprocessing* library. This allows for the model to have just acceleration data values as input unlike in other methods where the knowledge about machine parameters are required as input. This input is then used to train a single hidden layer Feed Forward Neural Network (FFNN) to train the model. The number of units in the hidden layer are fixed at *11* after performing some preliminary experiments that indicate that *11* units were enough to capture the non-linearity in the data. The max number of epochs are fixed at *100*. The number of units in the input layer (N_{input}) depends on the time interval (t_i) used in the time window (t_{window}) as follows.

$$N_{\text{input}} = t_{\text{window}} / t_{\text{i}} \tag{8}$$

The output layer as *1* unit which gives the probability of chatter occurrence. The hidden layer has ReLU activation and output layer has sigmoid activation. Stochastic gradient descent optimiser is used with the loss method being binary cross entropy. The default learning rate of *0.01* and momentum of *0* was chosen as safe values for all models. The architecture of the used NN is shown in Figure 5.

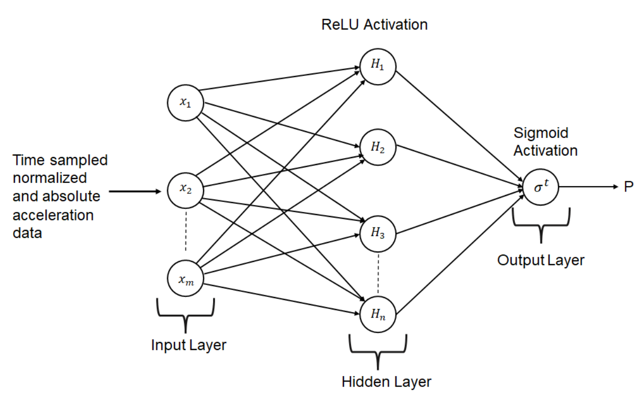


Fig. 5. Neural Network Architecture

The model with trained weights is then applied on real-time data by storing a buffer of acceleration values sampled for the time window and at the frequency specified for the

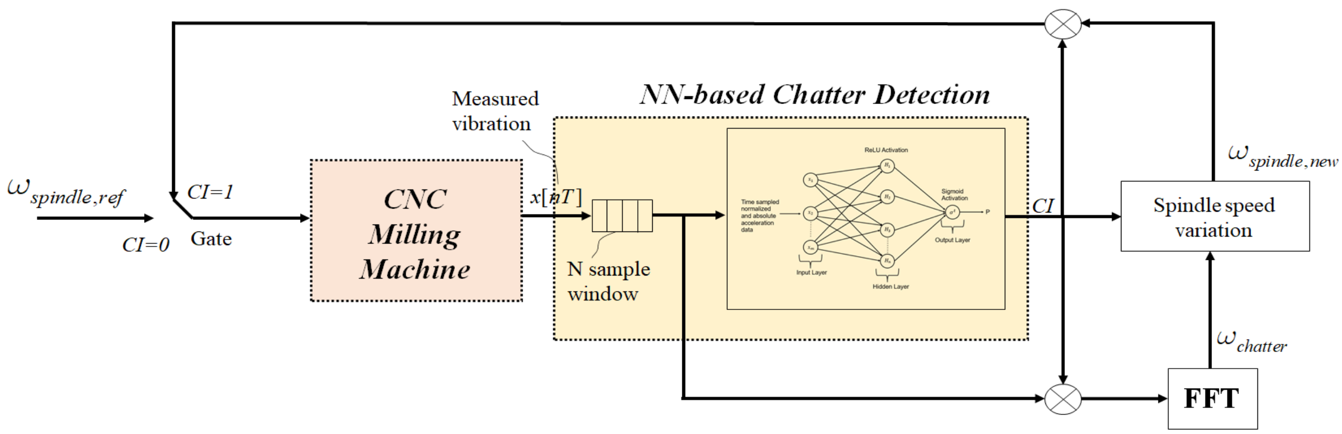


Fig. 6. Proposed chatter detection and control algorithm

model that is used. Then the absolute and standardized time sampled acceleration data will be used as input for the model. The output of the model will then be the probability of chatter occurrence. The model will predict the early onset of chatter depending on the gap with which the used model was trained for.

D. Chatter Control

The previous subsections of methodology have been used for chatter detection. After the chatter is detected with the aid of a neural network, a control loop should be implemented to suppress chatter vibrations. However, a real time experiment should be conducted to implement a control system, but it may create destructive results for the workpiece, tool and machine elements. In addition, the simulation used to generate training data does not allow to vary spindle speed. Therefore, the simulation needs to be improved to implement spindle speed variation control loop. Since that is a complex task, the control scheme is not implemented but presented theoretically in this report.

The overall block diagram can be seen in Figure 6 which includes chatter detection and control schemes. Once the NN is trained over simulations, it can be used to detect chatter. As can be seen in Figure 6, a reference spindle speed is set at first and process is started with chatter indicator *(CI=0)* which yields gate to be at reference spindle speed. Once chatter is detected, the indicator becomes *1*, and the FFT of last windowed data is generated. Chatter frequency is obtained between spindle rotation frequency harmonics as can be observed from Figure 3)c), and new spindle speed is adjusted by ensuring one of its harmonics to be at chatter frequency so that current chatter frequency energies become at spindle rotation frequency harmonics. At this time, the gate is fed with new spindle speed since the indicator is *1*.

IV. Experimental Setup

A. Correlation with classification variable and effect of parameters

The model is supposed to predict the early onset of chatter due to the concept of gap introduced in the previous section. The significance of the time sampled data with different values of gap in predicting the onset of chatter can be found with the correlation between the acceleration values and label output. Since the acceleration data is numerical and classification label is a categorical variable, point bi serial correlation method is used to compute significance of input features in predicting chatter detection.

Here, the correlation of input features for processed data with label output is calculated. The correlation of features with the label for a particular data set is averaged. To study the effect of time interval and time gap, this process is repeated for data sampled at different time intervals and with different time gaps. The value does not hold significance in absolute terms but the mean of correlation values for a data set can give relative importance of the time sampled data in predicting onset of chatter. It serves as a mode of comparison for significance of data sampled at different frequencies and with different time gaps.

Parameters	t_{gap} = 0 sec	t_{gap} = 0.2 sec	t_{gap} = 0.4 sec
t_i = 5 milliseconds	0.0466	0.0409	0.0463
t_i = 10 milliseconds	0.0456	0.0391	0.0581

TABLE I

Mean correlation of features with chatter occurrence

Table I shows the mean correlation of features with chatter occurrence for data sampled at different intervals and with different time gaps. It can be seen that mean correlation is higher with t_{gap} = *0.4 seconds* for data sampled at t_i = *10 ms*, indicating that the data sampled with the time gap *0.4s* has higher significance in predicting onset of chatter.

To study the actual effect of time interval and time gap in predicting the onset of chatter, six different models of FFNN were trained, one each for the combination of t_{gap} = *(0, 0.2, 0.4) s* and t_i = *(5, 10) ms*. The processed data from the simulation is split in the ratio *70:30* for training and validation to observe the performance on trained and unseen data of models trained on different data sets. The training progress for each model is tracked by running *10* iterations of each model and saving the accuracy for each run vs the number of epochs. The average accuracy for all iterations vs the number of epochs is then plotted for train and validation data sets. The score for best run for each model is also presented in the next section. Here, best score has been defined as the maximum of summation of accuracy on train set and test set at the end of training (*100* epochs) amongst all iterations.

B. Real-time Experimental Setup

In this report, *3* different acceleration data gathered from real-time milling chatter experiments are used. The data are taken from Manufacturing Process Control Laboratory at Oregon State University. The dataset contains acceleration versus time data and approximate chatter time. All experiments are conducted at *10 kHz* sampling rate, and various spindle speeds. Since the machining conditions and machine parameters are not supported, they are not presented in this paper. However, since the detection algorithm proposed in this paper is not machine specific, it is expected to work on real-time experiments that sense vibration regardless of spindle speed, depth of cut, or machine dynamics. The obtained acceleration data can be seen in Figure 7.

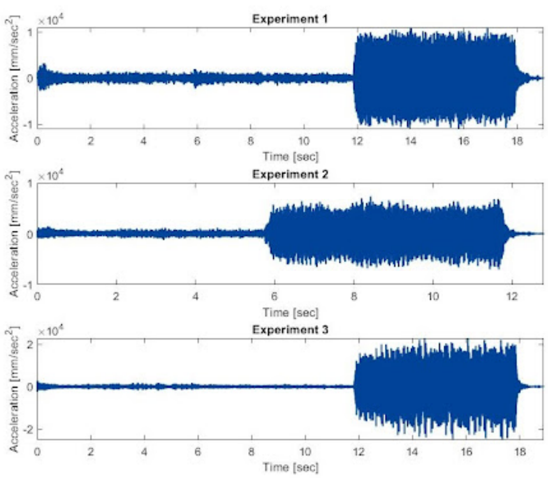


Fig. 7. Real time experiments

As can be seen, experiments start with stable cutting, and acceleration data jumps at around *12*, *6*, and *12* seconds respectively for experiment *1*, *2* and *3*.

V. Results

A. Neural Network Performance

1) Performance on training and validation sets: Figures 8)a), 8)b) and 8)c) show the training progress for models trained with data sampled at *10 ms* and Figure 8)d), 8)e) and 8)f) show the training progress for models trained with data sampled at *5 ms* for different time gaps.

It can be observed that increasing the time sampling frequency i.e training with data values sampled at lower time interval requires lesser epochs for the network to be trained. Another observation that can be made is that the the models trained on data with t_{gap} = *0.4 seconds* has a higher accuracy on validation data and less variance between train and validation data set accuracies than other specified time gaps. This is expected as it was seen in Table I that the model trained on data with t_{gap} = *0.4 seconds* had higher correlation with the classification output. Table II and III give the best model scores for model sampled at t_i = *5 and 10 ms* respectively for different time gaps.

Accuracy	t_{gap} = 0 sec	t_{gap} = 0.2 sec	t_{gap} = 0.4 sec
Train	94.6%	92.2%	93.5%
Validation	89.3%	88.4%	94.2%

TABLE II

Best model scores for time interval 10 milliseconds

Accuracy	t_{gap} = 0 sec	t_{gap} = 0.2 sec	t_{gap} = 0.4 sec
Train	96.2%	96.6%	95.2%
Validation	88.4%	84.4%	92.0%

TABLE III

Best model scores for time interval 5 milliseconds

It can be seen that modeling with this approach gives better results in predicting the onset of chatter earlier by *0.4 seconds*. The accuracy on validation set for early chatter detection is even better than spontaneous chatter detection. This has huge implications on the operation as early detection of chatter will allow preventive measures like spindle speed variation to be taken timely.

2) Performance on real time experimental data: Table IV and V give the predicted chatter times for models trained with different time gaps sampled at time intervals of *10* and *5 ms* respectively. The predicted chatter times are the times from $t = 0$ to the time model first predicts the onset of chatter in real time.

Time (secs)	t_{gap} = 0 sec	t_{gap} = 0.2 sec	t_{gap} = 0.4 sec
Experiment 1	12.04	12.09	12.17
Experiment 2	6.04	6.12	5.98
Experiment 3	12.11	12.18	12.10

TABLE IV

Predicted chatter time for real time experiments for time interval 10 milliseconds

Time (secs)	t_{gap} = 0 sec	t_{gap} = 0.2 sec	t_{gap} = 0.4 sec
Experiment 1	12.21	12.23	12.13
Experiment 2	6.12	6.11	6.09
Experiment 3	12.12	12.23	12.11

TABLE V

Predicted chatter time for real time experiments for time interval 5 milliseconds

It can be observed that models trained on data with time gap of *0.4s* do not exactly predict onset of chatter *0.4s* before the model trained with no time gap on data from

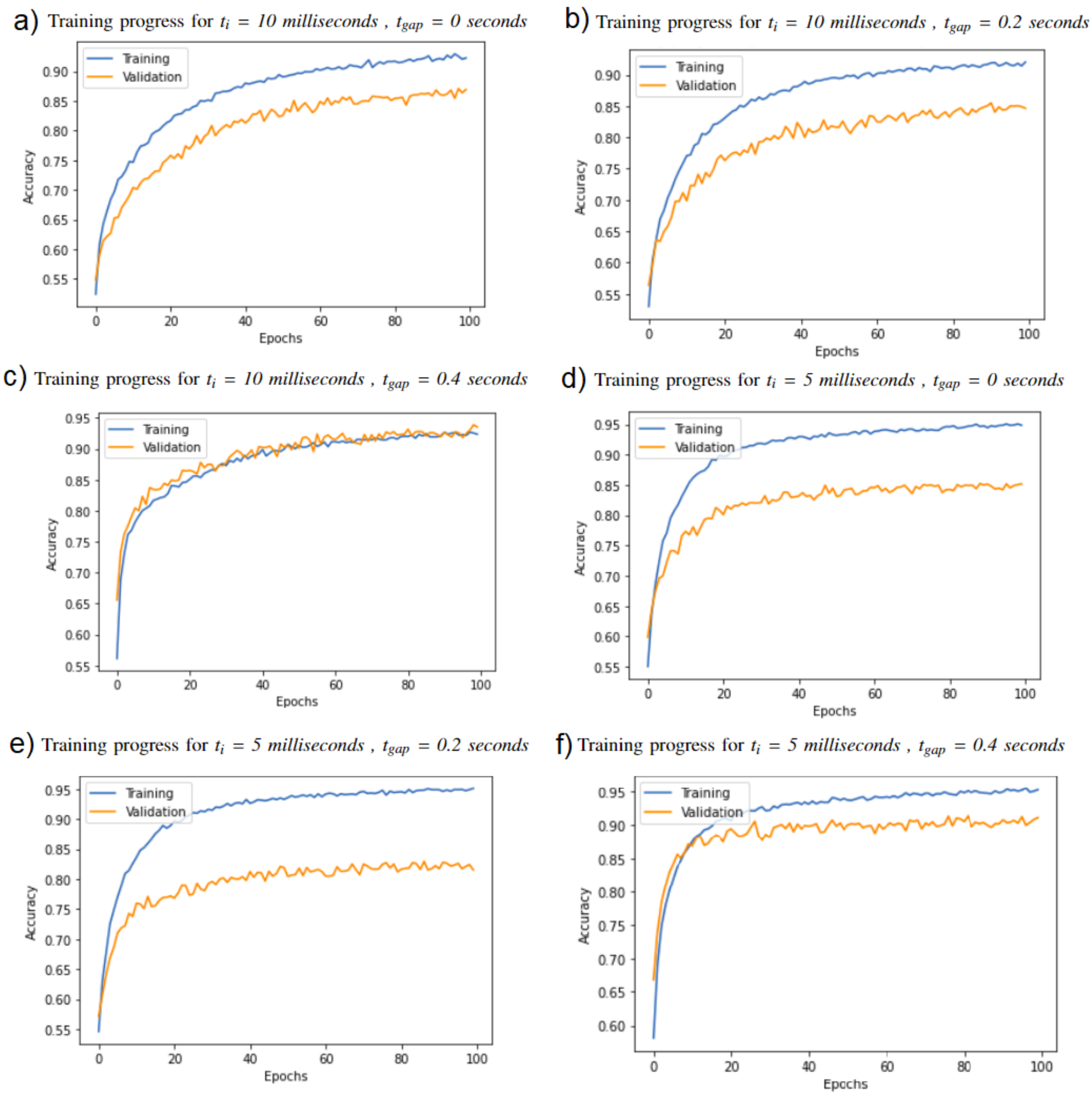


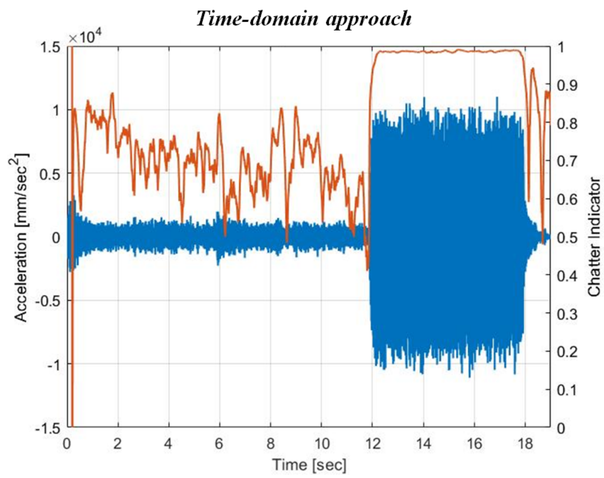
Fig. 8. Training progress by mean model accuracy over 10 iterations

real experiments. However, in most of the cases the models trained with time gap of *0.4s* still predict the onset of chatter few milliseconds before other models.

B. Comparison with other algorithms

The results given in Table IV and V are compared with two different algorithms from the literature. First, a time domain method in [7] is adapted to have less parameters. The cutting related energy is computed by modeling a Kalman Filter to track energy at spindle rotation frequency harmonics, and total energy is calculated by variance operator of the windowed signal as given in [13]. The chatter occurrence is detected once the energy ratio between chatter energy over total energy exceeds *0.5*. As the second method, a frequency domain approach is used as in [13]. Here, total energy is used again with variance, and the cutting related part is extracted by performing FFT at spindle rotation frequency harmonics. The algorithm detects presence of chatter once the ratio crosses the standardized threshold for detection of *0.5*. Both methods are performed on Experiment 1 given in Figure 7. The results given in Figure 9 are compared with Table IV and V.

As can be seen in Figure 9, time-domain method crosses *0.5* multiple times in stable cutting *(0-12 seconds)*. This is because the parameters of Kalman filter are not tuned for this machine and application, but are taken from the



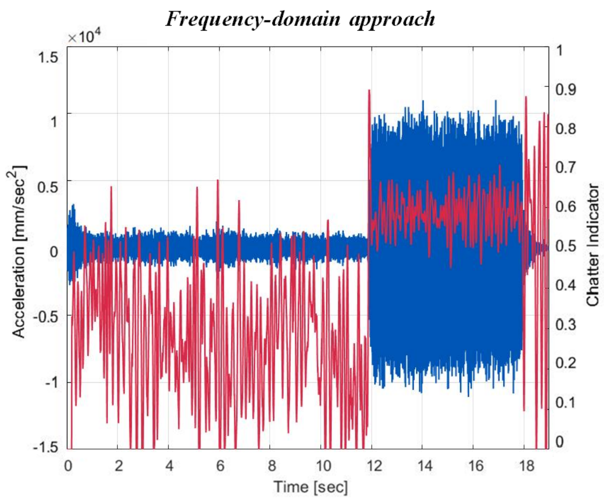


Fig. 9. Results of other algorithms on Experiment 1

reference paper [7] directly. Since sensor noise levels and covariances are different, the algorithm fails for this kind of application. Also, for frequency-domain method, there are multiple false alarms as well. This is because the window size is not sufficient to perform FFT on that. There are leakage issues, missing resolutions, and therefore, some of cutting related energy acts as chatter related which increases the ratio. On the other hand, the NN based method gives its alarm at as given in Table IV and V. This proves the robustness of the proposed algorithm. The advantages are as assumed before: **(i)** lesser parameters than time domain approaches (only window size to be set), **(ii)** not need to keep high resolution as in frequency domain approaches (trained over time-domain acceleration signal).

VI. Conclusion

The approach for chatter detection presented in this paper was able to predict the onset of chatter for real time experimental data without the use of machine tuning parameters due to it's use of normalized and absolute acceleration data values as input to the NN. This will result in the use of this approach for on-line chatter detection without the need to find machine tuning parameters which are specific for every machine. This will save time and cost associated with finding tuning parameters through experiments.

The concept of time gap was introduced for the early detection of onset of chatter. The performances on the processed simulation data indicated that the models trained with time gap of *0.4s* have a higher accuracy on unseen validation data. Although on the real time experimental data these models did not exactly predict the onset of chatter before the specified time gap yet they still predicted the onset of chatter few milliseconds before the model trained with no time gap. This can have huge implication on machining operations where such a prediction in advance gives the operator the ability of using techniques like spindle speed variation to prevent the damages due to chattering.

VII. Future Work

The complete algorithm given in Figure 6 will be performed on real-time chatter control by improving the simulation to have spindle speed variation. Moreover, it will be performed on real experimental setup. Once the chatter detection part is completed, a control loop will be introduced based on spindle speed variation. It is proposed that the data in the up most current window is used to have frequency spectrum by Fourier Transform (FT). Then, by adjusting a new spindle speed that includes the current chatter frequency as a harmonic of spindle rotation frequency, chatter is suppressed.

In this approach a FFNN has been used and the time gaps have been manually selected for early chatter onset detection. Due to the process of manual selection of time gaps, an accurate way of determining which time gap will lead to data that is more correlated to the classification output could not be found. In future, a Recurrent Neural Network (RNN) will be used to train the model to detect early onset of chatter. Since RNN handles sequential data in a much more elegant manner, the need to manually select the time gap may be eliminated. Also, the flow of information is better controlled using an RNN therefore only meaningful data will be used for the detection of onset of chatter.

References

[1] Tony L Schmitz and K Scott Smith. *Machining dynamics*. Springer, 2014.

[2] Ronald Faassen. Chatter prediction and control for high-speed milling. *Eindhoven: Eindhoven University of Technology*, 362, 2007.

[3] Ding Chen, Xiaojian Zhang, Huan Zhao, and Han Ding. Development of a novel online chatter monitoring system for flexible milling process. *Mechanical Systems and Signal Processing*, 159:107799, 2021.

[4] Yusuf Altintas. *Manufacturing automation: metal cutting mechanics, machine tool vibrations, and CNC design*. Cambridge university press, 2012.

[5] Yusuf Altintaş and Erhan Budak. Analytical prediction of stability lobes in milling. *CIRP annals*, 44(1):357–362, 1995.

[6] Christian Brecher, Prateek Chavan, and Alexander Epple. Efficient determination of stability lobe diagrams by in-process varying of spindle speed and cutting depth. *Advances in Manufacturing*, 6(3):272–279, 2018.

[7] Hakan Caliskan, Zekai Murat Kilic, and Yusuf Altintas. On-line energy-based milling chatter detection. *Journal of Manufacturing Science and Engineering*, 140(11):111012, 2018.

[8] Jiri Tlusty. The stability of the machine tool against self-excited vibration in machining. *Proc. Int. Res. in Production Engineering, Pittsburgh, ASME*, 465, 1963.

[9] Emad Al-Regib and Jun Ni. Chatter detection in machining using nonlinear energy operator. *Journal of dynamic systems, measurement, and control*, 132(3), 2010.

[10] MQ Tran and MK Liu. Chatter identification in end milling process based on cutting force signal processing. In *IOP Conference Series: Materials Science and Engineering*, volume 654, page 012001. IOP Publishing, 2019.

[11] MF Zaeh, F Schnoes, B Obst, and D Hartmann. Combined offline simulation and online adaptation approach for the accuracy improvement of milling robots. *CIRP Annals*, 69(1):337–340, 2020.

[12] NJM Van Dijk, EJJ Doppenberg, RPH Faassen, N Van De Wouw, JAJ Oosterling, and H Nijmeijer. Automatic in-process chatter avoidance in the high-speed milling process. *Journal of dynamic systems, measurement, and control*, 132(3), 2010.

[13] Ryo Koike. *Realtime monitoring and stability diagnosis of cutting process by applying disturbance observer*. PhD thesis, Keio University, 2016.

[14] M Hossein Rahimi, Hoai Nam Huynh, and Yusuf Altintas. On-line chatter detection in milling with hybrid machine learning and physics-based model. *CIRP Journal of Manufacturing Science and Technology*, 35:25–40, 2021.

[15] Mourad Lamraoui, Mustapha Barakat, Marc Thomas, and Mohamed El Badaoui. Chatter detection in milling machines by neural network classification and feature selection. *Journal of Vibration and Control*, 21(7):1251–1266, 2015.

Work Distribution

Organization

Satyam Bhawsinghka 70%
Mert Ozdemir 30%

Technical

Mert Ozdemir 50%
Kaan Bahtiyar 50%

Coding

Satyam Bhawsinghka 50%
Kaan Bahtiyar 50%

Writing

Kaan Bahtiyar 33%
Satyam Bhawsinghka 33%
Mert Ozdemir 33%